Call-A.P.P.L.E.

May 1984
Vol. 7, No. 5

Worlds Largest Apple User Group

$3.00
$3.75 Canada
£2.00 UK

ROBOTICS

- HERO I + Apple II = ?
- Robots Go To School
- Artificial Intelligence: Myth and Reality
- Robot Control Languages Like Talking to a Friend
- Building and Running Your Own Mechanical Helper

A First Look at Apple's NEW IIc

- A.P.P.L.E.'s Exclusive Preview
- First Photos
- 65C02 Microprocessor
- Scribe Printer
  B&W and Color for under $300

INSIDE:
New Products
Write—A.P.P.L.E.
A.P.P.L.E. Doctor

HARDWARE/ SOFTWARE REVIEWS
- Apple][Mouse
- Gibson Light Pen

TECHNICAL
- Pros and Cons of ProDOS

PLUS
Useful Information from Apple Graphics to Spreadsheets

Apple PugetSound Program Library Exchange

Glide through a full side of programs each month.

EVERY MONTH IS DIFFERENT!

All of these programs were on a single recent issue. . .

Poker Solitaire
Play several interlocking poker hands at once.
Concentration
Match the squares, like the popular TV show.
Twenty Easy Pieces
A hi-res jigsaw puzzle
Five-Dicer
Similar to the popular Yahtzee game
Diet Program
Weight Schedules based on calories, activity, weight, etc.
Fact Sheet
A friendly Rolodex filing system
Cosmic Rebound
Lo-res graphics novelty
Copy +
An improved version of Apple's own COPYA

Doctor Dave
An educational program to teach the parts of speech
Gas Law Problems
An educational program for chemistry
Flip-Spin-Twisties
How to dress-up your screen with the flipping batons (a la Beagle Brothers)
General Ledger Tutorial — Part One
Write your own general ledger systems
Random Access Files Tutorial
How to build your own disk files
DOS Tutorial
The inner workings of DOS

Q&D Track Copier, Q&D File Overlap Checker, and Q&D Disc Scanner
Invaluable recovery aids for those times when a diskette "goes bad"
Fatcat
Allows you to increase the maximum number of files allowed in your catalog.
As if that weren't enough . . .
This SOFTDISK issue also includes graphics, music, miscellaneous programming hints and tips and even the program listings from SOFTALK magazine!

SOFTDISK

3811 St. Vincent, Shreveport, LA 71108 318-868-7247

SOFTDISK comes on double-sided diskettes. It is a bonanza of unprotected games, tutorials, utilities, educational software, and articles. You keep one disk and return the other after copying it. We pay the postage and furnish a mailer. What's more, you become part of the action when you send us comments and programs on the disk you return.

"Great in the classroom, too!"
APPLESOFT DOS 3.3

I HAVE ENCLOSED:
☐ $10 for a single issue.

☐ $69.95 for a year of SOFTDISK.

Name _____

Address _____

City/State _____ ZIP_____

Visa/Mastercard # _____

Exp. _____

Circle 1 on Reader Service Card

Pick a Number Between pfs®:file and dBASE II®....

You've outgrown the simple file management program you've been using. Perhaps you need more than one diskette of data in a file, or greater math capabilities, or password file protection, or a more powerful report generator. You know which program the hot-shot programmer down the street uses, but you still think that "multiple relations" refers to your in-laws, and you're not sure you *want* to know "query language". Besides, if it's so easy to use, why all those expensive training seminars and books and programs to make it easier? Don't trade in your computer on a bunch of used file cabinets. . . .

Check out the alternatives . . .

ADVANCED DB MASTER

The most powerful user-oriented information management system available for the IBM PC®, PC/XT®, 100% compatibles, and SAGE™ II/IV computers. Works with hard disk or floppies. 256K RAM required.

- Up to 44 diskettes of data per file!
- Built-in letter writing/mail merge capability
- Exchange data with 1-2-3™, VisiCalc®, Multiplan™, etc.
- Password file protection
- Keystroke macros

Retails for $595.

Try Before You Buy!

For a working sample version of Advanced DB Master, see your Stoneware dealer, or send $9.95 (refundable with purchase) to Sample Version, Stoneware, Inc., 50 Belvedere St., San Rafael, CA 94901.

DB MASTER 4 PLUS

We integrated all the functions* of Utility Paks #1 and #2 into DB Master Version Four and created DB Master 4 Plus. Now you get more features than ever for Apple® II+/IIe or Franklin Ace™ 1000/1200 information management—**with no increase in price!**

- Up to 100 or more diskettes of data per file!
- VisiFile™ and PFS file converters
- File restructure capability
- DIF™ translators
- Global editor (including date math)
- Label printer up to 5 across)
- Damaged file recovery

*(except Transaction File Merge)

Retails for $350.

Free Brochure

For a free brochure describing DB Master 4 Plus, see your local Stoneware dealer.

Power Without Programming!

STONEWARE®

STONEWARE INCORPORATED • 50 BELVEDERE STREET • SAN RAFAEL, CA 94901
TELEPHONE (415) 454-6500 • TELEX II: 910-384-2016 STONEWARE SRFL

REGISTERED TRADEMARKS:
Stoneware (Stoneware, Incorporated) • IBM PC and IBM PC/XT (International Business Machines Corporation) • dBASE II (Ashton-Tate) • pfs (Software Publishing Corporation) • VisiCalc (VisiCorp)

TRADEMARKS:
Advanced DB Master and DB Master (DB Master Associates) • Apple II+ and IIe (Apple Computer, Inc.) • DIF (Software Arts Corp.) • Franklin ACE (Franklin Computer Corporation) • 1-2-3 (Lotus Development Corporation) • Multiplan (Microsoft Corporation) • VisiFile (VisiCorp)

Circle 29 on Reader Service Card

Apple PugetSound Program Library Exchange
21246 - 68th Ave. S.
Kent, WA 98032 - (206) 872-2245
Hours: 9am to 4pm PST, Mon-Fri.

Officers and Directors
Dick Hubert President
Bob Huelsdonk Vice President
Dave Lingwood Secretary
Bob Clardy Treasurer
Dave DeGroot Director
Val J. Golding Director
Don Williams Director

Administration
Dick Hubert Executive Director
Don Steele Asst to Executive Director
Dave O'Leary Controller
Sally Davies Manager of Purchasing
Steve Johnson Technical Staff
Charles Stillman Manager of Education Services

Departments
Order Desk
Tracing
Member Information
Education, Classes
Retail store
Dealer services
Marketing
Technical support
Testing and Evaluation
Operations

Publications
Gary Kruger Executive Publisher
Val J. Golding Editor
Kathryn Hallgrimson Suther Associate Editor
Donald Elman Editorial Coordinator
Kaiulani Schuler Editorial Operations
Ed Adams Art Director
Ron Keister Cover Artist
Paul Sleeper Pascal Editor
Grawin Publications Advertising (206) 223-0861
John Elliott Advertising Coordinator
Johathan Post Contributing Editor

Staff writers
Cornelis Bongers
Wes Felty
Jim Hockenhull
Clifton M. Howard, M.D.
Ken Kashmarek
Paul Mosher
Bill Parker
Lee Reynolds
Clay Ruth
Michael Shadick
David G. Sparks
Rick Sutcliffe
Ralph H. Swerdlow, M.D.
Lee Webster

Member Services

24 hour toll-free order line:
1-800-426-3667
(except Washington state)
Orders ONLY
Information not available

Programming/Technical Assistance
Call-A.P.P.L.E. HOTLINE
(206) 872-9004
9am to 3pm - 6pm to 10pm
(Pacific time)
7 days per week

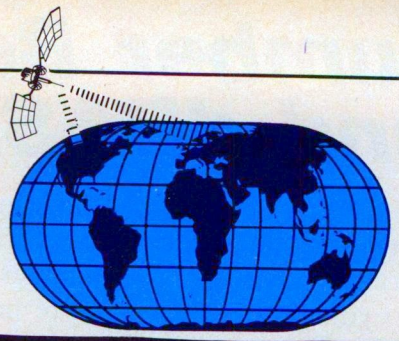
Consultants
see listings on tear-off card inside

A.P.P.L.E. Crate Bulletin Board
(206) 872-6784

Meetings
7:30pm third Tuesday of each month
A.P.P.L.E. Office
21246 68th Ave. S., Kent
(except as noted)

Membership Information
is available on postpaid tear-off card inside

A.P.P.L.E. on the SOURCE
A special section of The Source is open only to A.P.P.L.E. members who may download Call-A.P.P.L.E. programs. To access, type: CALL-A.P.P.L.E. from Source command mode. To register, send mail to ST1570 with your name, membership number, expiration date and Source account number.
Printed in U.S.A.



Call-

New From Apple

7 **A First Look at Apple's NEW *IIc***
Dave Lingwood
A.P.P.L.E.'s exclusive preview of Apple's newest computer for everyone, with a look at a marvelous new color printer.

Robotics

10 **About Robots: An Account of Costs**
Michael Thyng
A recipe for building a mechanical friend, including parts and prices.

13 **HEROs and Apples**
John Covington and Burt Webb
How to control a Heathkit HERO-1 robot with an Apple computer.

17 **Artificial Intelligence: The Myth and the Reality**
Jim Wicker, M.A.
Many hope (or fear) computers will someday think like humans. Is that day here?

19 **Using an Apple to Speak English to a Robot**
Susan K. Lynch, RB Robot Corp.
Would you rather talk to robots in BASIC or English? A new language for the RB5X gives you a choice.

Graphics

23 **The Graphics Toolkit: Part 2**
Randi J. Rost
More graphic revelations about Apple II graphics commands.

27 **Hardware/Software Review — The Gibson Light Pen: An "Etch-a-Sketch" Graphics System for the Apple**
Lee Webster
A "point-by-point" discussion of an easy-to-use input device and its extensive software package.

Hardware

34 **Hardware Review — The AppleMouse *II***
John Cox
That adorable electronic rodent is now ready to do its thing for the Apple II.

ADVERTISER INDEX

Access Micro. 38
Artemis Systems, Inc. 25
Beagle Bros 40, 41
Bytes & Pieces. 66
Central Point Software, Inc. 60
Datamost C-3
David Data 28
Douglas Electronics 72
Graysoft. 45
Hardcore Computist 31
Holly Advertising 39
Hollywood Hardware. 79
Laing Electronics 59 63
Lazerware. 21
Micro-Analyst, Inc. 50
Name Brand Software. 69
Nibble Magazine. 74
Nibble Notch. 49
Quorum International Unltd. . . . 33
Sibsource Inc. 11
Softdisk. C-2
Softcell Inc. 18
Softronics. 5

A.P.P.L.E.™

Apple PugetSound Program Library Exchange

39 **Book Review — The Apple // Circuit Description**
Leonard H. Anderson
A circuit tracer's delight.

Technical

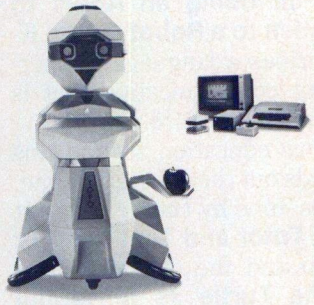
42 **Adding a TYPE Command to ProDOS**
Cecil Fretwell
The expandability of the ProDOS/BASIC.SYSTEM is demonstrated by adding a utility command.

51 **ProDOS — Pros and Cons**
Cornelis Bongers
A balanced view of ProDOS, with a comparison between the BASIS 108 and Apple IIe computers.

Business

55 **THE SPREADSHEET SHEET: Data Analysis (Part 6)**
Jeffrey M. Jacques, Ph.D.
This month's installment shows how inferential statistics can be performed with the spreadsheet.

Education



61 **TOPO — A Tutor and Tool for Your Classroom**
Harvey Barnett
A tangible form of Logo turtle, TOPO helps students refine their computer skills.

62 **The Turtle as a Helpful Robot**
Nola Sheffer
Young children learn abstract concepts with a Logo-controlled device.

65 **Touch Sensors for Screen Turtles**
Jeff Sandys
These routines make your Logo screen turtle actually bounce off walls.

Letters

68 **A.P.P.L.E. Doctor**
Bruce F. Field
Another fringe benefit of A.P.P.L.E. membership—cures for your computer ills.

70 **Write—A.P.P.L.E.**
THE READERS

Departments

4 **The INNER CIRCUIT**
Don Elman
An editorial survey of this month's feature articles.

22 **Member Billboard**
Don Elman
How A.P.P.L.E. strives to provide the best possible service to its members.

78 **Tomorrow's Apples Today**
New product announcements from Apple Computer, Inc. and other companies.

80 **Unclassifieds**

Apple user groups may reprint without prior permission any portion of the contents herein, provided author, title, and publication credits are given and a written notice of such use furnished to the editorial office. For commercial applications of programs herein, the author should be contacted c/o General Office. Call-A.P.P.L.E. is published monthly by Apple PugetSound Library Exchange, 21214 6-68th Ave. S. Kent, WA 98032. Subscriptions are not available but copies are mailed free as a benefit of A.P.P.L.E. membership. Subscriptions only can be purchased for $21 per year.

Entire contents copyright © 1984 by Apple PugetSound Library Exchange.

Software Banc..... 46
Southeastern Software..... 26
StarLogic..... 54
Stellation Two..... 16
Stoneware..... 1
Street Electronics..... 8
Strobe Inc..... 12
Telephone Software Connection..... 33
Titan Technologies..... C-4
Troy Software Design, Inc. 35
Userfest..... 76, 77
VF Associates..... 52

Changes of address must be received at the General Office at least 30 days prior to effective date of move. Issues missed due to non-receipt of change of address notification must be separately reordered by the member at current cost. Change of address form may be found on this page.

I am moving and my present address is:

PRESENT:

NAME _____

ADDRESS _____

CITY _____

STATE _____ ZIP CODE _____

EFFECTIVE DATE _____

NEW ADDRESS:

NAME _____

ADDRESS _____

CITY _____

STATE _____ ZIP CODE _____

MEMBERSHIP NUMBER _____

* Please fill out and return this notice six weeks before moving.

The INNER CIRCUIT

Don Elman

If there ever were any doubts (heaven forbid) that Call -A.P.P.L.E. is on the cutting edge of the newest technologies and products relating to Apple computers, then this issue will certainly put them to rest. Thanks to the wonderful efforts of our outgoing editors, Val Golding and Kathryn Hallgrimson Suther (whose influence, incidentally, will be found in these pages for many months to come), we begin with an excellent series on robotics. In **About Robots: An Account of Costs**, Michael Thyng leads off with a recipe for building your own robot, complete with prices, part descriptions, instructions, and outside resources. John Covington and Burt Webb describe a more-than-mathematical link between the HERO 1 and the Apple *II* in **HEROs and Apples**.

As any science fiction fan knows, of course, "true" robots must be able to think independently. The prospects for it ever occurring are discussed by John Wicker in **Artificial Intelligence: The Myth and the Reality**. A first step in that direction might be the ability to communicate with robots in a language that is more natural to humans than to computers. In **Using an Apple to Speak English to a Robot**, Susan K. Lynch describes a language for person-to-robot communication that is eerily similar to standard English.

Our entire education section this month rounds out the robotics theme. Two articles—one by Harvey Barnett (**TOPO—A Tutor and Tool for Your Classroom**) and the other by Nola Sheffer (**The Turtle as a Helpful Robot**)—show how programmable robot turtles become the living embodiment of Logo. For more advanced Logo programmers, Jeff Sandys explains how a tactile sense can be imparted to the moving icon in **Touch Sensors for Screen Turtles**.

Also in this issue are features on graphics, several hardware and software reviews, another installment in Jeffrey Jacques' Spreadsheet series, a technical focus on ProDOS, and such popular features as **A.P.P.L.E. Doctor** and **Write—A.P.P.L.E.** Surely this is a feast of delights that could satisfy nearly anyone's tastes.

But if that isn't enough, we present as the icing on the cake **A First Look at Apple's NEW *IIc***, a detailed overview of the newest Apple computer. Even though it wasn't publicly announced until after press time, A.P.P.L.E. officer Dave Lingwood, accompanied by Dick Hubert and Gary Kruger, were treated to a special, advanced showing of the *IIc* down at Apple headquarters just days earlier. Not only does Dave tell us about the *IIc* itself in all its glorious specifications, he also describes an amazing, inexpensive color printer that Apple is introducing along with the new computer, and gives some technical details about the 65C02 microprocessor which is at the *IIc*'s core. Thanks, Dave, for such an excellent and speedy job of reporting!

With such an embarrassment of riches in this issue, you may be wondering what we could possibly do next month to top it. Well, just keep listening for the sweet melodies of computerized music off in the distance, and tune in next time to find out.

Letter from the President

Dear A.P.P.L.E. Members,

In tribute to Val Golding, we acknowledge the end of an era at A.P.P.L.E. Sadly, the May issue of *Call -A.P.P.L.E.* marks the end of the editorship of our beloved and revered founding member, Val J. Golding. Seemingly, one of the constant factors affecting our club is change, and this particular change is one that leaves all of us at A.P.P.L.E. missing a special person, who made so many significant contributions to the club we enjoy today.

For newer members, perhaps a brief history is in order. A.P.P.L.E. began in Val's basement. Although at the inception Val had other employment responsibilities, his enthusiasm and zeal to share Apple computing created a second full-time position, that of administering the then infant-sized A.P.P.L.E. Val did it all in those days—hotline, orders, mailings. His interest and dedication served as inspiration to all of us. Seven years later, not to mention some 20,000 members, he will be sorely missed.

Val has now made his own fine entry into the publishing arena. His new publication, *The Apple's Apprentice*, is already on sale and doing well. This new magazine is the very first dedicated exclusively to children who use Apples, and all of us at A.P.P.L.E. wish him the great success he deserves.

In the meantime, there will be no shortage of the journalistic excellence readers of *Call-A.P.P.L.E.* have come to know. Our new editorial coordinator, Donald Elman, is now at the helm. Don comes to us with a fine publishing background, and is sure to continue the tradition of technical superiority and excellence with which Val's departure leaves us.

We know Val will be watching closely with care and help. Fortunately for us all, he is still on our Board of Directors, and will always be there for the input only he can deliver.

Good luck, Val!

Dick

Dick Hubert
President

NOW YOUR APPLE® CAN HOOK TO ANY MAINFRAME.

Apple is a registered trademark of Apple Computer, Inc.,
CP/M is a registered trademark of Digital Research, Inc.,
Dow Jones News/Retrieval is a registered trademark of
Dow Jones, Inc. The Source is a service mark of
Source Telecomputing Corporation,
CompuServe is a registered
trademark of CompuServe, Inc.

If you have an Apple II, Apple II Plus, or Apple IIe, we have good news for you. Now there are two inexpensive software programs that can turn your Apple into a much more valuable tool.

Softerm 1 connects you with information services.
Softerm 1 lets you retrieve information from services such as *The Source,*℠ *CompuServe®,* and *Dow Jones News/Retrieval®.* Plus, gives you the ability to access bulletin boards and send or receive electronic mail. Other features include user-defined keyboard macros, built-in phone book for automatic dialing, terminal mode line capture simultaneously to print or disk, copy screen to print or disk, and terminal status display.

Softerm 2 connects you with your company's computer.
This expanded version of Softerm lets you gain access to the information stored in your company's main computer from your home or office. With either version of Softerm, you can down load information into your Apple and capture it on your own disk in any format you choose—DOS, CP/M®, or Pascal. Also included with Softerm is a source program for your host computer to ensure compatibility with Softerm's file transfer capabilities.

Makes your Apple work exactly like any major terminal.
Softerm 2 provides complete emulation of these terminals: ADDS Regent 20, 25, 40, 60; ADDS Viewpoint; Data General D200; Datapoint 3601; DEC VT102, VT52; Hazeltine 1400, 1410, 1500, 1520; Hewlett-Packard 2622A; Honeywell VIP7205; VIP7801, VIP7803; IBM 3101 Model 10 and 20; Lear Siegler ADM-3A, ADM-5 and TeleVideo 910, 925, 950. And the list is growing all the time. We'll send you a User's Guide, handy reference card, and a telephone number to call if you need more assistance.

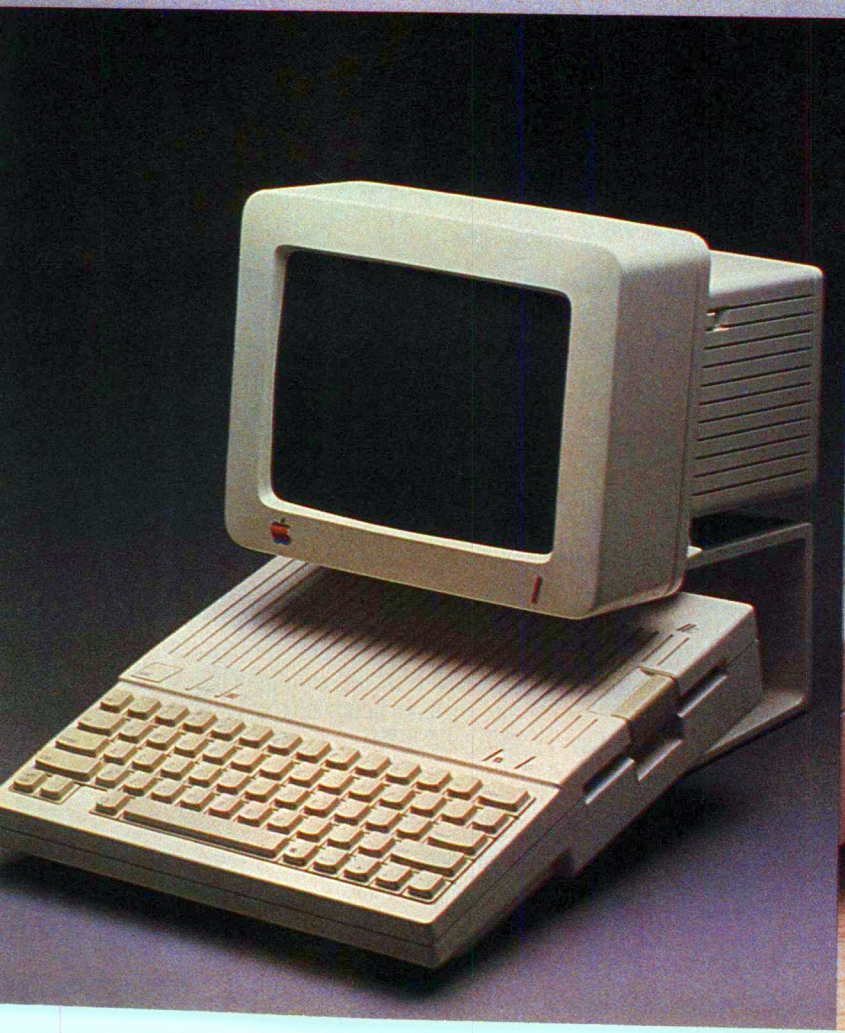
Softerm can't become obsolete.
We constantly improve Softerm. And we make those improvements available to you 24 hours a day, 7 days a week. To update your program, just dial our computer and transfer the improvements to your disk.

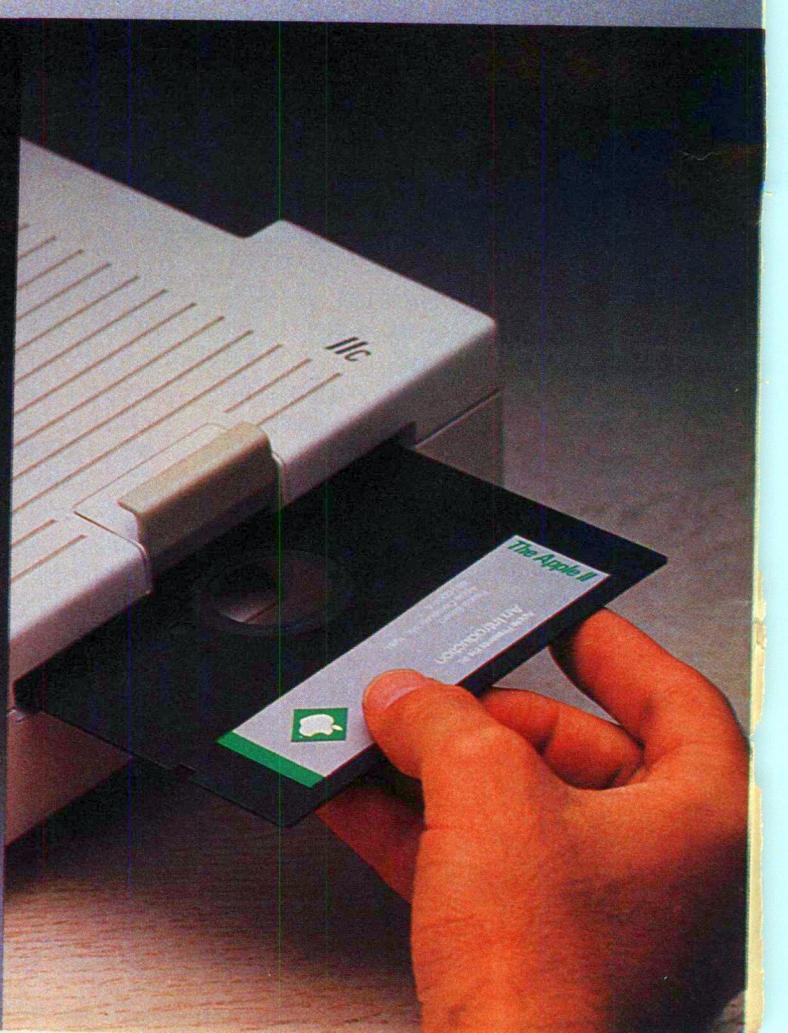
Get Softerm now.
It's the inexpensive way to turn your Apple into a much more valuable tool for your home or office. Softerm 1 is only $135; Softerm 2 is $195. Both are available now from your local dealer or from Softronics, Inc. For more information or to order, call (901) 683-6850.

SOFTRONICS
3639 New Getwell Road, Suite 10
Memphis, TN 38118. (901) 683-6850

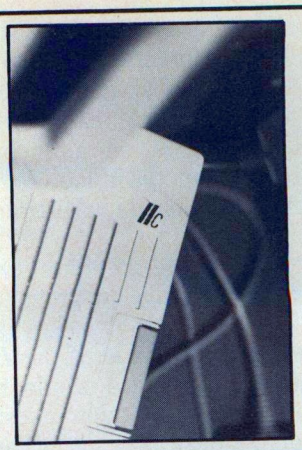
Circle 37 on Reader Service Card

The Apple II
Software Registration Card
The Apple II




IIc
The Apple II

A First Look at Apple's New IIc



Dave Lingwood

It is here. Apple's portable version of the *IIe*, called the *IIc*. Why *IIc*? "C" stands for the 65C02 CMOS processor chip it uses, a compatible update of the familiar 6502. The processor isn't really the big news; it is the packaging that counts.

Basically, the Apple *IIc* is a "slotless" 128K *IIe*, with one built-in drive, in a 7½ lb. package.

Apple Computer introduced it to the world April 24th, along with a range of new hardware and software. They bought out San Francisco's Moscone Center for the affair, but it happened after our press deadline for this issue. The details here come from a pre-announcement meeting the folks at Apple gave us.

This month we'll concentrate on the *IIc* and its marketing, and on new Apple peripherals. Next month we'll look at the S.F. doings, and see what external hardware and software developers have already done for this new machine.

THE ROLE OF A NEW MACHINE

The *IIc* isn't just a new computer—it is a total package of computer, peripherals, documentation and introductory software. It's a fresh, red-Apple colored, upbeat theme designed to attract people new to computers.

Apple designed all of this with their eye definitely on the novice "high end" personal user — they make it simple to set up and use without having to add any extras. An RF modulator and TV connector are included. The manuals and diskettes provide a good introductory course in computer literacy.

The photos tell the story. The *IIc* certainly doesn't look like any other Apple product. It has a very European look. The computer and peripherals all carry the fresh, clean "stamp" of the professional designer (the people who designed Sony's Walkman, in fact). The *IIc* generates lots of "c" words in your mind: compact, carryable, capable, and CUTE! This styling is the look of the future for other products and peripherals in the *II* family.

The *IIc* fills a niche in the 6502 family line. The Apple *III* is still at the top of the line, and fits more closely in it due to the SOS compatibility of ProDOS — the operating system supplied with the *IIc*.

Next in the family pecking order is a full-blown professional package *IIe*, at $2,995, with two floppy drives, 128K, a ProFile hard disk, and the Catalyst *IIe* software package (from Quark Engineering).

Surprisingly, the *IIc* with its $1,300 price tag is next in the hierarchy, not the bottom of the line. This price buys you the red-Apple bright box containing the *IIc*, external AC power pack, RF modulator and TV connector, manuals, and introductory diskettes. Everything you need to plug in and go.

The lowest-cost machine is a 40-column 64K *IIe* with one drive, for a new price of $995.

Don't bet that there won't be other members of the family in the future. The theme of the San Francisco announcement event was "The Apple *II* Forever". They mean it.

OUTSIDE

The *IIc* looks slim and European — as if Phillips or Norelco designed it. The ivory-colored, rectangular case, with rounded corners, is a mere 12 x 11½ x 2½ inches in size. The tan, short-throw keys have a layout identical to the *IIe*, but with a lighter touch and a tactile click. Three DIN connectors on the crammed back panel reinforce the feeling. The machine itself weighs 7½ lbs. A fold out handle on the back lets you carry it in a keyboard-down position, and prop up the back end for use, giving the proper keyboard angle and better convection cooling.

The AC power supply is external, bringing only +15vdc to the small DC-to-DC power supply inside the computer. It attaches to one of the rear panel DIN connectors by 6 feet of cable. There is an equal length of AC cable on the other end of the supply. If you want to take off for the hills, the *IIc* can be run from a 12 volt source. This should start quite a run, in the short term, on 12 volt monitors. The *IIc* will run off of any DC source from 9 to 18 volts. If the power drops below nine volts, the power-on LED will begin to blink.

The single 5¼ inch drive is built in, on the right side of the computer. It is an ALPS unit, inaudible while spinning. It pops the diskette out when you open the door. The disk drive LED is in the case, viewable through a slot located above the DELETE key. We tested the drive with a protected word processor that requires a truly Apple-compatible drive. The program booted perfectly.

The reset key is above the 1 key, flush with the case. It is set up to require CONTROL-RESET, like later model][+s and *IIe* machines. Next to it are two pen-pushable switches. One is for 40/80 columns. If it is pushed down before the machine is turned on, software can sense it ($C060 > $7F) and switch to 40 to match the user's TV set. Use the up position for a monitor. The other pen-push switch reconfigures the keyboard to Dvorak style! With the

appropriate character generator ROM, this switch lets *IIc* have a choice between English and French, Spanish, etc. It is one reason why Apple is making a simultaneous release of the computer in four European countries, Canada, Australia, and the U.S.

Above the delete key are two LEDs: red for disk drive on, green for power on.

The rectangular shape of the *IIc* means that the keyboard would be parallel to the desk top if you just sat the thing on the desk. The carry handle folds down to prop up the back to get the proper keyboard slope. The keys had a somewhat "looser" feel than on my early-model *IIe*, but I had the impression that my touch typing would soon adjust to the unit.

There is an external speaker jack on the right, and a speaker volume control. Plug your Walkman into the jack and the internal speaker shuts off.

The back panel is crowded. Each of the connectors is marked with unmistakable "icons" that tell you what plugs in where. Different connector designs make sure you can't make a mistake in plugging things in.

There is a mouse connector on the back (DB9), and built-in circuitry to drive it. Paddles or joy-stick will also plug in here. An "icon" character generator ROM chip is used. This has some graphics-like characters that are used by the mouse firmware. These icons are located in one of the two code ranges where inverse upper case is on the *IIe*.

Also on the back panel is a DB15 RGB output similar to that on the Apple*III*. The RF modulator also connects here. There is also one of the remarkably non-standard DB19 disk connectors, into which the optional second drive connects.

There is a standard video output phono jack.

By Fall, there will be an 80 x 24 LCD flat panel display that hooks onto the DB15 connector. It is a compact package, about 10 x 6 x 1 inches. This isn't here yet, however. In the meantime, a standard monitor will work.

While there are no slots inside, the most commonly used "boards" are built in: disk controller in six, 80 columns with 64K auxiliary memory in three, a Super Serial for your printer in slot one location, and an identical Super Serial in slot two location for modem use. Both of these exit through DIN connectors on the back. Why no parallel connector? Apple has standardized on serial printers. The reason, as we hear it, is that RF emissions requirements could not be met if the Apple had a parallel ribbon that hangs *20 or so* potential antenna wires out the back. We'd be surprised if serial-to-parallel converters don't start popping up on the market from other vendors.

INSIDE

On booting, the screen comes up "Apple *IIc*," but everything else works as on the *IIe* — with some minor improvements. The nicest is that commands need not be typed in upper case. There are supposedly some editing enhancements, but we didn't have time to test them.

I quickly forgot that the computer isn't a *IIe*. Test programs did what they were supposed to do. Of course, you have your choice ProDOS, or your own DOS 3.3 or Pascal diskettes. ProDOS gives you pseudo-disk access to the extra 64K through the "/RAM" path.

CALL-151 drops you into the monitor as usual. Disassembling in the monitor you'll see ROM code in some unexpected places (e.g., the monitor JSRs to $CCCC at one point). The "slot" code in the $C1-$CF pages is there as you'd expect with Super Serial, auxiliary, and slot 6 controller cards. Most noticeable are the new 65C02 op codes. For example PHY PHX PLY BRA (branch always), etc. — see the accompanying article. There are more such codes used in the $C1-$CF space than in the rest of the monitor, where it seems that less extensive patching was done. Caution: if you have software that uses monitor entry points different than those supported on the *IIe*, you are quite likely to branch to East Nowhere.

Technies take note: if you open the *IIc* case your warranty is voided! The manuals stress that there is nothing you can do inside. However...

Remove six bottom screws and the case comes apart, though not as simply as does the][+ or *IIe*. The screws bite into nothing more than plastic; we wouldn't try opening it once a day. Inside are just about the same number of chips as in a 128K *IIe*, no surprise — until you

A.P.P.L.E. CONSULTANTS

John A. Hall, Jr.	Apple IIe (East)	(617) 241-9050
Cecil Fretwell	Apple IIe (MidWest)	(319) 236-0961
Bob Huelsdonk	Apple III (West)	(206) 362-4910
Harry T. Hanson	Apple III (East)	(201) 467-0712
Carmen Picardi	Apple III (East)	(603) 895-4447
Stuart Yusem	Apple III (East)	(609) 822-9306
Tim Smith	Apple III (MidWest)	(389) 734-7451
Neil Clemmons	Agriculture	(314) 968-9618
Erv Edge	Assembly (Alaska)	(907) 337-7620
Dennis Baumgarte	Assembly (East)	(914) 298-8756
Murray Arnow	CPM (MidWest)	(312) 867-8000
Kevin Talbot	CPM (West)	(206) 232-7058
Steven E. Blair	Data Bases	(913) 782-5901
Jerry Rogan	Data Bases, Business	(319) 332-4166
William P. Steinberg	DOS (East)	(212) 921-0663
Art Schumer	DOS (North)	(701) 282-7907
Lee Meador	DOS (South)	(817) 469-6019
Wm. Reynolds III	DOS (West)	(503) 472-4028
Ted Perry	Education	(916) 485-1690
Cornelis Bongers	DOS, Applesoft (Europe)	010-525511 Ext. 3275
Alan Robinson	Fortran	(503) 757-8234
Jim Eatherly	Games	(202) 232-6046
Tedd Sperling	Geoscience	(517) 393-1296
Stan Axelrod	Hardware Design/I-O Control	(415) 563-1840
Tom Deal, M.D.	Handicapped Applications	(206) 634-2847
Randy Halvorson	Hardware Interfacing	(206) 783-9040
Ed So	Hi-Res	(408) 267-3448
Stowe Keller	Mathematics	(804) 979-4619
Fred Merchant, M.D.	Medical Group	(415) 921-7792
Stephen F. Lee	Music	(704) 289-4102
Tom Woteki	Pascal (East)	(202) 547-0984
Tim Smith	Pascal (MidWest)	(309) 734-7451
Paul E. Sleeper	Pascal (West)	(415) 494-1915
Bruce E. Sargent	Pascal (West)	(206) 488-1161
Dennis R. Cohen	Pascal (West)	(213) 956-8559
Earl Keyser	Pilot Language	(612) 429-5570
Randy Warrens	Pilot Language	(608) 837-9566
Jeff Sandys	Logo (Western U.S.)	(206) 784-3317
Hersch Pilloff	Stock Market	(301) 292-3100
Jim Hockenhull	Video Interfacing	(509) 334-4137
George L. Hefter	VisiCalc (East)	(617) 256-3082
Vince Aderente	VisiCalc (North)	(312) 289-9685
Dr. Bob Richard	VisiCalc (South)	(504) 466-0894
Brian Abrahams	VisiCalc (South Africa)	(Durban) 315442
Michael Ginsberg	Word Processing	(805) 735-8210

Call-A.P.P.L.E. consultants are unpaid volunteers and perform a valuable service. Before placing a call, please make sure you know the time zone you are calling to. No calls after 9pm, please.

We are always looking for more consultants. If you see an area that needs coverage, please drop us a note in the mail with your qualifications and availability.

ORDER PRODUCTS LISTED ON REVERSE

Name _____ Membership # _____ Expiration date _____ Date _____

Address _____ City _____ State/Country _____ Zip _____ Phone _____

$3.00 Handling charge on all orders under $25.00

	CODE	PRODUCT	SHIPPING WT.	UNIT PRICE	AMOUNT
1.					
2.					
3.					
4.					
5.					
6.					

Total Shipping Weight _____ SUBTOTAL _____

Total Shipping Cost _____

Washington residents add 8.1% sales tax _____

$3.00 Handling charge if order is under $25.00 _____

TOTAL AMOUNT _____

PLEASE USE PRODUCT NAME AND CODE WHEN ORDERING

VISA Account No. _____ Exp. Date _____

MasterCard Account No. _____ Exp. Date _____

Mail to:
A.P.P.L.E.
21246 68th Ave. S.
Kent, WA 98032
(206) 872-2245

FOREIGN MEMBERS: Please calculate postage based on weights in product descriptions. (See postage table below for cost.)

Sorry, No Foreign Hardware Orders

FOREIGN SHIPPING CHART

POSTAL TABLE	Zone I West Indies Central America, Columbia, Venezuela	Zone II All South American Countries (except in Table I), Europe, N. Africa	Zone III Asia, Mid East, S. Africa Australia and Pacific Islands
0 - 4 oz.	$1.28	$ 1.68	$ 2.08
4 - 8 oz.	1.80	2.60	3.40
8 - 12 oz.	2.32	3.53	4.72
12-16 oz.	2.84	4.44	6.04
16-20 oz.	3.36	5.36	7.36
20-24 oz.	3.88	6.28	8.68
24-32 oz.	4.92	8.12	11.32
2 - 3 lb.	7.00	11.80	16.60
3 - 4 lb.	9.08	15.48	21.88
Each additional ½ lb. over 4 pounds add:	1.04	1.84	2.64

CANADA SHIPPING RATES

0-4 oz.	$ .71
4-8 oz.	1.39
8-12 oz.	2.07
12-16 oz.	2.58
2 lb.	3.10
3 lb.	4.10
4 lb.	5.10

For weights over 4 lbs: $5.10 + $1.00 additional per pound.

UPS 2ND DAY AIR

WEIGHT NOT TO EXCEED	48 STATES A	HAWAII D
1 lb.	$3.00	$4.53
2 "	4.00	5.71
3 "	5.00	6.89
4 "	6.00	8.07
5 "	6.50	9.25
6 "	7.50	10.43
7 "	8.50	11.61
8 "	9.50	12.79
9 "	10.50	13.97
10 "	11.50	15.15
11 "	12.50	16.33
12 "	13.50	17.51
13 "	14.50	18.69
14 "	15.50	19.87
15 "	16.50	21.05
16 "	17.50	22.23
17 "	18.50	23.41
18 "	19.50	24.59
19 "	20.00	25.77
20 "	21.00	26.95
21 "	22.00	28.13
22 "	23.00	29.31
23 "	24.00	30.49
24 "	25.00	31.67
25 "	26.00	32.85

PLEASE SPECIFY WHEN ORDERING:

☐ APPLE II ☐ LISA ☐ APPLE IIc
☐ APPLE IIe ☐ MACINTOSH ☐ APPLE II+
☐ APPLE III

A.P.P.L.E.™ PRODUCTS

AVAILABLE TO A.P.P.L.E MEMBERS ONLY

SOFTWARE

☐	SAMP	AMPERMANAGER	$22.50
☐	SAPF	AP FILE	$18.50
☐	SFLR	AP GRAPHICS "The Filler"	$17.00
☐	SAM 3	APMAIL II VERSION 3.0	$25.00
☐	SAPMPR	APMAIL.Pro	$35.00
☐	SAPMUP	AP MAIL UPDATE (Incl.Ser.#)	$10.00
☐	SAPT	AP TEST	$18.50
☐	SCAR	APPLESOFT CARPENTER//	$18.50
☐	SMLC	BIG MAC LC	$28.50
☐	SDISRE	LIFESAVER By Microlab	$21.00
☐	SDAN	DRIVE ANALYZER	$32.00
☐	SEAP	E Z PILOT	$28.50
☐	SDIC	DICE	$12.50
☐	STITWR	DIRECTORY TITLE WRITER	$15.00
☐	SD6B	UTILITY DISKPAK 6B-16	$18.50
☐	SHFI	HIGHER FONTS 1	$13.00
☐	SHTP	HIGHER TEXT PLUS	$27.50
☐	SIMAGI	IMAGINATOR	$60.00
☐	SMUL	MULTI-TRIEVE	$50.00
☐	SPGE	PROGRAM GLOBAL EDITOR	$18.50
☐	SPLE	PROGRAM LINE EDITOR	$15.00
☐	SPZAP	PROZAP	$25.00
☐	SS70	SOFT SEVENTY	$20.00
☐	SSYM	SYMBOL SYMON	$20.00
☐	STSS	THE SPREADSHEET	$75.00
☐	STSC	THE SPREADSHEET 1.0 to 2.0 data file converter	$5.00
☐	STUR	TURTLESOFT	$22.50
		ValueSoft	
☐	SCATKI	CATER KILLER	$12.50
☐	SDOSAN	DISK ANALYZER 1.3	$12.50
☐	SFARWO	FARWORD	$12.50
☐	SGRAFM	GRAFMAGIC	$12.50
☐	SGRABU	GRAPHICS BUILDER	$12.50
☐	SMICWT	MICROWRITER //e	$12.50
☐	SMORTG	MORTAGES, DEPREC. AND FINANCIAL CALCULATIONS	$12.50
☐	SSCRAM	SCRAMBLER	$12.50
☐	SGRADE	SCHOOLHOUSE GRADER	$12.50
☐	SZAPT	ZAPT	$12.50
☐	SWICWA	WICKERWARE #1	$12.50

APPLE /// PRODUCTS

☐	SWW3	WORD WEAVER ///	$75.00
☐	HDR3	APPLE /// DISK DRIVE	$285.00

HARDWARE

☐	HSRMON	AMBER SCREEN MONITOR By Leading Edge	$100.00
☐	HAPD	AP DRIVE	$235.00
☐	HAPC	AP CONTROLLER	$85.00
☐	HAKK	AP COOL KIT	$20.00
☐	HLC1	AP L CASE 1	$30.00
☐	HLC2	AP L CASE 2 DICE	$25.00
☐	HLC1A	w/ AP L CASE 1	$37.50
☐	HLC2B	w/ AP L CASE 2	$30.00
☐	H80C	APPLE //e 80 COLUMN CARD	$120.00
☐	HAUD	APPLESURANCE by Prometheus	$85.00
☐	HP16	AP RAM 16 K by Prometheus	$50.00
☐	HP28	AP RAM 128 K by Prometheus	$295.00
☐	HMPC	DUMPLING GX	$130.00
☐	HMPD	DUMPLING 64	$180.00
		GRAPHITTI by Prometheus	$85.00
		Please specify printer type:	
☐	HGRAEP	Epson, Panasonic	
☐	HGRACI	C. Itoh	
☐	HGRAOK	Okidata Prowriter, Microline, NEC	
☐	HGRAID	IDS (460, 560, Prism)	
☐	HGRAAP	Apple Dot Matrix	
☐	HP11	HOBBY PROTOTYPING BOARD	$21.00
☐	HHOC	HOME CONTROLLER by Prometheus	$40.00
☐	HPAD	KOALA PAD TOUCH TABLET	$95.00
☐	HJOYST	MACH II JOYSTICK By Hayes	$34.00
☐	HJMACH	MACH III JOYSTICK By Hayes	$44.00
☐	HJOMA3	MACH III JOYSTICK By Hayes (for //e)	$49.00
☐	HMOD	MODEM II By Multi-Tech	$195.00
☐	HPANMO	PANASONIC CT-160 COLOR MONITOR	$279.00
☐	HPANPR	PANASONIC KX1090 PRINTER	$325.00
☐	HSERIN	RS232 SERIAL INTERFACE CARD	$85.00
☐	HPANRI	PANASONIC P1090 RIBBONS	$14.00
☐	HPADDL	PADDLE-ADAPPLE By SCRG	$25.00
☐	HPRMOD	PROMODEM 1200	$450.00
☐	HPRT	PROWRITER PRINTER	$380.00
☐	HPRC	Prowriter Ribbon cartridges	$5.00
☐	HRIGNO	RIGHTNOTCH	$15.00
☐	HRPT	REPEATERRR	$23.00
☐	HSOFTC	SOFTCASE "FLASHCARD" By Synetix	$39.00
☐	HSSS	144 K SINGLE DRIVE	$275.00
☐	HSSU	144 K UPGRADE KIT	$200.00
☐	HSSD	288 K DUAL DRIVE	$500.00
☐	HFAN	SUPER FAN II Zener Ray	$79.00
☐	HTRC	TALLY MT 160L RIBBON	$12.50
☐	HTRI	TALLY MT 180L RIBBON	$17.00
☐	HULTRA	THE ULTRATERM By Videx	$310.00
☐	HTR120	TRANSTAR 120 Letter Quality Printer	$559.00
☐	HSUPMO	SUP R MOD By M&R	$60.00
☐	HURB	ULTRA ROM BOARD/EDITOR Firmware Enhancement System	$149.00
☐	HVRC1	VERSACARD by Prometheus with 6' par. cable	$150.00
☐	HVRC2M	with 6' male ser. cable	$150.00
☐	HVRC2F	with 6' female ser. cable	$150.00
☐	HVRC3	with 1' ser. cable	$150.00
☐	HVSLM	6' male ser. cable only	$20.00
☐	HVSLF	6' female ser. cable only	$20.00
☐	HVSS	1' ser. cable only	$20.00
		VERSACARD graphics PROMs	$30.00
☐	HVPREP	Epson, Panasonic	
☐	HVPRCI	C. Itoh	
☐	HVPRID	IDS	
☐	HVPROK	Okidata, ProWriter, NEC	
☐	HVPRAP	Apple D.M.	
☐	HVIE80	VIEWMASTER 80 COLUMN CARD By Applied Engineering	$149.00†

A.P.P.L.E. GAMES

☐	SKOALA	KOALA GRAMS SPELLING I	$31.50
☐	SSPI	SPIDER EATER	$24.00
☐	SCCO	COSMIC COMBAT	$12.50
☐	SD7	DISKPAK 7	$17.50
☐	SMCU	MUMMY'S CURSE	$12.50
☐	STHB	THUNDERBOMBS	$16.00
☐	SMIC	MICROBE	$30.00

A.P.P.L.E. LIBRARY

☐	PBAD	BENEATH APPLE DOS Call-A.P.P.L.E. IN DEPTH	$17.00
☐	PID1	VOL I ALL ABOUT APPLESOFT	$7.00
☐	SIN1	VOL I Diskette	$16.50
☐	PID2	VOL II ALL ABOUT PASCAL	$11.00
☐	SIN2	VOL II Diskettes	$20.00
☐	PID3	VOL III ALL ABOUT DOS	$13.50
☐	SIN3	VOL III Diskettes	$15.00
		Call-A.P.P.L.E. BACK ISSUES	
☐	MM-A 80	Call-A.P.P.L.E March/April 80	$2.00
☐	MMay 80	Call-A.P.P.L.E May 80	$2.00
☐	MJun 80	Call-A.P.P.L.E June 80	$2.00
☐	MN-D 80	Call-A.P.P.L.E Nov/Dec 80	$2.00
☐	MJan 81	Call-A.P.P.L.E January 81	$2.50
☐	MM-A 81	Call-A.P.P.L.E March/April 81	$2.50
☐	MJun 81	Call-A.P.P.L.E June 81	$2.50
☐	MJ-A 81	Call-A.P.P.L.E July/August 81	$2.50
☐	MSept 81	Call-A.P.P.L.E September 81	$2.50
☐	MOct 81	Call-A.P.P.L.E October 81	$2.50
☐	MN-D 81	Call-A.P.P.L.E Nov/Dec 81	$2.50
☐	MJan 82	Call-A.P.P.L.E January 82	$2.50
☐	MFeb 82	Call-A.P.P.L.E February 82	$2.50
☐	MMar 82	Call-A.P.P.L.E March 82	$2.50
☐	MApr 82	Call-A.P.P.L.E April 82	$2.50
☐	MMay 82	Call-A.P.P.L.E May 82	$2.50
☐	MJun 82	Call-A.P.P.L.E June 82	$2.50
☐	MJul 82	Call-A.P.P.L.E July 82	$2.50
☐	MAug 82	Call-A.P.P.L.E August 82	$2.50
☐	MSept 82	Call-A.P.P.L.E September 82	$2.50
☐	MOct 82	Call-A.P.P.L.E October 82	$2.50
☐	MNov 82	Call-A.P.P.L.E November 82	$2.50
☐	MDec 82	Call-A.P.P.L.E December 82	$2.50
☐	MJan 83	Call-A.P.P.L.E January 83	$2.50
☐	MFeb 83	Call-A.P.P.L.E February 83	$2.50
☐	MMar 83	Call-A.P.P.L.E March 83	$2.50
☐	MApr 83	Call-A.P.P.L.E April 83	$2.50
☐	MMay 83	Call-A.P.P.L.E May 83	$2.50
☐	MJun 83	Call-A.P.P.L.E June 83	$2.50
☐	MJul 83	Call-A.P.P.L.E July 83	$2.50
☐	MAug 83	Call-A.P.P.L.E August 83	$2.50
☐	MSept 83	Call-A.P.P.L.E September 83	$2.50
☐	MOct 83	Call-A.P.P.L.E October 83	$2.50
☐	MNov 83	Call-A.P.P.L.E November 83	$2.50
☐	MDec 83	Call-A.P.P.L.E December 83	$2.50
☐	MJan 84	Call-A.P.P.L.E January 84	$3.00
☐	MFeb 84	Call-A.P.P.L.E February 84	$3.00
☐	MMar 84	Call-A.P.P.L.E March 84	$3.00
☐	MApr 84	Call-A.P.P.L.E April 84	$3.00
☐	MMay 84	Call-A.P.P.L.E May 84	$3.00

CALL-A.P.P.L.E. DISK OF THE MONTH

☐	SCA6	6 mos. (start 1/84)	$55.00
☐	SCA12	12 mos. (start 1/84)	$95.00
☐	SM0184	DISK OF THE MONTH 01/84 (JAN)	$11.00
☐	SM0284	DISK OF THE MONTH 02/84 (FEB)	$11.00
☐	SM0384	DISK OF THE MONTH 03/84 (MAR)	$11.00
☐	SM0484	DISK OF THE MONTH 04/84 (APR)	$11.00
☐	SM0584	DISK OF THE MONTH 05/84 (MAY)	$11.00
☐	SM0684	DISK OF THE MONTH 06/84 (JUN)	$11.00
☐	SM0784	DISK OF THE MONTH 07/84 (JUL)	$11.00
☐	SM0884	DISK OF THE MONTH 08/84 (AUG)	$11.00
☐	SM0984	DISK OF THE MONTH 09/84 (SEP)	$11.00
☐	SM1084	DISK OF THE MONTH 10/84 (OCT)	$11.00
☐	SM1184	DISK OF THE MONTH 11/84 (NOV)	$11.00
☐	SM1284	DISK OF THE MONTH 12/84 (DEC)	$11.00
☐	PPK2	VOL II 1979 Compendium	$10.00
☐	PPK3	VOL III 1980 Compendium	$15.00
☐	PPK4	BOUND SET (Call-A.P.P.L.E.) 1981	$18.50
☐	PPK5	BOUND SET (Call-A.P.P.L.E.) 1982	$18.50
☐	PPK6	BOUND SET (Call-A.P.P.L.E.)1983	$18.50
		PEEKing at Call-A.P.P.L.E.	
		ANTHOLOGY DISKETTES (each)	$12.50
☐	SA1	VOL 1 (Jan thru Jun 1979)	
☐	SA2	VOL 2 (Jul thru Dec 1979)	
☐	SA3	VOL 3 (Jan thru Jun 1980)	
☐	SA4	VOL 4 (Jul thru Dec 1980)	
☐	SA5	VOL 5 (Jan Thru Jun 1981)	
☐	SA6	VOL 6 (Jul thru Dec 1981)	
☐	SA7	VOL 7 (Jan thru Jun 1982)	
☐	SA8	VOL 8 (Jul thru Dec 1982)	
☐	SA9A	VOL 9a (Jan to March 1983)	
☐	SA9B	VOL 9b (March to June 1983)	
☐	SA10A	VOL 10a (Jul to Sept 1983)	
☐	SA10B	VOL 10B (Oct to Dec 1983)	
☐	SPA1	Pascal '81	
☐	PACD	THE APPLE II CIRCUIT DESCRIPTION	$18.50
		THE POWER OF VISICALC	
☐	PVIS	VOL I	$8.50
☐	PVI2	VOL II	$8.50
☐	PU65	USING 6502 ASSEMBLY LANGUAGE	$17.00
☐	PASL	ASSEMBLY LINES: THE BOOK	$17.00
		WHAT'S WHERE IN THE APPLE	
☐	PWWA	COMPLETE VOLUME	$17.00
☐	PWW2	THE GUIDE	$8.00
☐	PKID	KIDS AND THE APPLE	$16.00
☐	SPDS	PUBLIC DOMAIN SOFTWARE (each disk volume)	$5.00
☐	SEAMON	EAMON ADVENTURES	$5.00
☐	PELA	THE ELEMENTARY APPLE	$11.00
☐	PWTA	WOZ TALKS (VHS video tape)	$27.50
☐	PIMAGS	IMAGINATOR SOURCE LISTING	$30.00

A.P.P.L.E. SERVICES AND ACCESSORIES

☐	HPAG	TYPING AID	$13.00
☐	FSOU	THE SOURCE SUBSCRIPTION	$60.00
☐	HFLS	FLIPSORT	$22.50
☐	HDBA	DISK BANKS BLANK DISKETTES	$4.50
☐	HDUST	APPLE DUST COVER	$8.00
☐	HVEB	Verbatim Datalife (Box of 10)	$25.00
☐	HCDB	Control Data Corp (Box of 10)	$22.00
☐	HGEN	BULK BLANK PACK SS, DD, DISKS set of 100 with labels)	$150.00
☐	LID	LISA DISKETTES	$90.00
☐	HPOST	Call-A.P.P.L.E cover posters (specify month)	$9.95
☐	HTSH	T-Shirts ☐ S ☐ M ☐ L ☐ XL	$10.95
☐	FAPP	A.P.P.L.E. MEMBERSHIP ($25.00 first time application fee, $26.00 first years dues)	$51.00
☐	FREN26	A.P.P.L.E. MEMBERSHIP RENEWAL	$26.00
☐	FSPSVC	Special handling for rush orders	$25.00

SPECIALS

☐	SPANTH	1983 ANTHOLOGY SPECIAL (With $50 order or new membership) (expires 5/31/84)	$35.00

* Indicates Introductory Price
† Subject to Availability

Use Order Form on Reverse

remember that this unit contains the equivalent of two serial cards, controller, mouse, and auxiliary memory/80 columns. The disk controller is all on ONE chip, marked "IWM" — the "integrated Woz machine".

All chips are soldered in, except for three of the ROMs and the processor. From the looks of it, board level repairs will not happen — just pull off the DC–DC power supply and swap in a new mother board.

The keyboard is connected via the standard short ribbon cable. It is repairable, but most dealers will probably just swap if something goes wrong. A plastic membrane under the keys guides your spilled Coke down and out through the ventilation slots under the keyboard front. A rubber sheet under the membrane is there just to improve the feel of the keys.

How durable is it? After all, this is a portable machine. The electronics should survive normal bumping about even better than the][+ or *IIe*.

cont'd on p. 73

The 65C02 Microprocessor

An update of an old friend, the 65C02 uses CMOS design to lower power consumption and improve reliability through greater noise immunity. In the bargain you get an expanded set of op codes and two new addressing modes. All without mirrors, yet.

First, some design improvements: on the 65C02, all invalid op codes are treated as NOPs. This avoids hanging the processor, or invalid results.

The decimal flag is set to binary after resets and interrupts. On the 6502 it was indeterminate after a reset. Sloppy.

On the 6502, an indirect jump with an xxFF operand did not increment the page address. It does on the 65C02, adding one additional cycle to the instruction.

Op Codes

These are the goodies, from the standpoint of the programmer.

For example, how often have you needed to do an unconditional branch, for example in a loop, and have to stop to figure out which flag will always be set? How often have you goofed, and had your code fall out of the loop unexpectedly? No more. The 65C02 has a BRA or "branch relative always" instruction.

All information here was taken from the NCR 65C02 product description. GTE also makes this chip, as does Rockwell. There are some differences with the Rockwell unit, as we understand it, which makes it a bad fit with the Apple.

Here are the new op codes:

Hex	Mnemonic	Description
80	BRA	Branch relative always
3A	DEA	Decrement accumulator
1A	INA	Increment accumulator
DA	PHX	Push X register onto stack
5A	PHY	Push Y register onto stack
FA	PLX	Pull X register from stack
7A	PLY	Pull Y register from stack
9C	STZ	addr Store Zero value, absolute
9E	STZ	addr,x Store Zero, indirect indexed by X
64	STZ	z.p. Store Zero in zero page
74	STZ	z.p.,x Store Zero in zero page,X
1C	TRB	addr Test and reset memory bits with accumulator, absolute
14	TRB	z.p. Test and reset memory bits with accumulator, zero page
0C	TSB	addr Test and set memory bits with accumulator, absolute
04	TSB	z.p. Test and set memory bits with accumulator, zero page
89	BIT	# Test Immediate with accumulator

And these new addressing modes for existing commands:

72	ADC z.p.	Add memory to accumulator, zero page
32	AND z.p.	AND memory with accumulator, zero page
3C	BIT abs,X	Test memory with accumulator, absolute indexed by X
34	BIT z.p.,X	Test memory with ccumulator, zero page indexed by X
D2	CMP (z.p.)	Compare memory and accumulator, zero page indirect
52	EOR (z.p.)	Exclusive OR, zero page indirect
7C	JMP abs(X)	Jump absolute, indexed indirect by X (new addressing mode) X + addr LH = address of L byte of LH effective address
B2	LDA (z.p.)	Load accumulator, zero page indirect
12	ORA (z.p.)	OR accumulator with memory, zero page indirect
F2	SBC (z.p.)	Subtract memory from accumulator, zero page indirect
92	STA (z.p.)	Store accumulator, zero page indirect

"Zero page indirect" or "(z.p.)" is a new addressing mode. The second byte of the instruction points to a zero page location that contains the L byte of an LH byte pair that is the effective address. Handy.

The absolute indexed indirect by X JMP instruction will be very handy for jumps using a "jump table" containing several destination addresses. For example:

```
LDX #COMMAND
JMP TBL,(X)
TBL: DFB ADDR1
DFB ADDR2
DFB ADDR3
```

pick up a #2 if you want to jump to command address 2, do the jump, which adds X to the address of the table base, then finds the final JMP address bytes at (in this case) ADDR2. Whee!

The 65C02 will run at either one or two MHz. We know you can plug one into a *IIe* if you want. You can't use one on a][; the machine's timing is too slow and near the ragged edge. We don't know yet about the *III*.

About Robots: An Account of Costs

Michael Thyng

THIS time I want to tell you about the costs involved in building your own robot, the design I used, and where the problems of construction arose. Before we start, let me thank all of you who have been writing encouragement and telling me about your robot projects. Sounds like there's a *lot* of interest. I'm glad to hear it. The more of us there are doing independent research in robotics, the sooner the general public will realize there are alternatives to organic pets.

I started construction in July, 1982. At this time, I met with two technical advisors, neither providing the needed assistance. One didn't have any technical skill, the other just talked a lot. (You'll probably run into this problem also.) I've met a lot of people who say they want to be involved, but when you call on them to participate, they're too busy, or they just want to "learn" (which means any time you spend will be only one third as productive). I think most of us are almost *forced* to build these things alone. Before I leave this topic, please note: You have to talk to a lot of people. Out of maybe 50 people I talked with at various stages, perhaps only 14 of them had useful things to contribute. But, those 14 were worth the time wasted on the other 36. A word here, an idea there. It all adds up to success, if you listen.

My next step was to visit the library, finding reams of material. Unfortunately, most of it was quite dated. Many of the books were written in the fifties and have a "gosh won't this be amazing" attitude about robots. Your local electronic supply houses are good sources for robotics information. Many of them have at least three books on the subject, and that's enough to get you started. Another source is your local bookstore. They stay in business because they sell what's hot, and robots are hot. They may have good robotics information for you, even if they don't carry a single book with the word "robot" in the title. Look for titles having to do with digital electronics, books on microprocessor interfacing, and also digital applications. Concerning interfacing, look for authors with names like Rodnay Zaks, Christopher or Jonathan Titus, David Larsen, Don Lancaster, Lance Leventhal, and Leo J. Scanlon.

I have a certain loyalty to Lance Leventhal's *Microcomputer Experimentation with the MOS Technology KIM-1,* since my robot — MIRT — is guided by a KIM-1 microcomputer. Another author who used the KIM-1 and wrote about it is Tod Loofbourrow in *How to Build a Computer Controlled Robot.* Frank DaCosta was the first author I read who told how to build your own microcomputer board using an 8085 processor. He includes complete schematics and machine language routines. His book is entitled *How to Build your own Working Robot Pet.* Frank's book helped me get prepared to build my own microcomputer on a perf board and build a robot pet.

I would be totally unqualified to talk about robots if I hadn't read at least one of David L. Heiserman's books — I think he's completed four. He turned my thinking around with a couple of paragraphs in *How to Design & Build Your Own Custom Robot.* I quoted him before in this column, but the wisdom of his words are worth repeating. "I have never lost the impression that such machines can, and should be, regarded as pets and companions."

That impression colors all my thinking about robots and machine intelligence in general. Unlike so many other experimenters, I have a hard time thinking about these machines as potential slaves that are programmed to do my bidding. I am convinced that thinking about machines as slaves clamps an impenetratable lid on an experimenter's imagination. It seems to straightjacket one's thinking.

Just play this little game in your head for a couple of weeks. Force yourself to think of robots as pets and companions, instead of slaves. Dream about a machine that can tell you what *it* wants to do. Censor any thoughts about your playing a direct role in dictating the machine's behavior.

As you work through this exercise, you will find your impressions of robots blossoming into new and exciting dimensions. Don't let "reality" mess up your dreaming! Let yourself go and you'll begin painting a remarkable mental image. Practical "reality" will come crashing into your dreams later on, but that's a terrible place to start.

Two other Heiserman books worth reading are *Build your own Working Robot* and *Robot Intelligence with Experiments.* You should at least read the last one if you are really interested in robots.

Back to construction. I kept a log for ten months. During that time I recorded 60 hours of activities. But that doesn't count those six hour marathons toward the end of construction to bring the project to an end. I think 100 hours would be a fair estimate of how long it takes to build a robot. Then there's the matter of education. I bought all the books listed previously, and then some. I also spent a lot of time in libraries

reading the material I could find. So, I've got maybe 1000 hours into the project, with only 10 percent of the time directly attributable to the robot's construction.

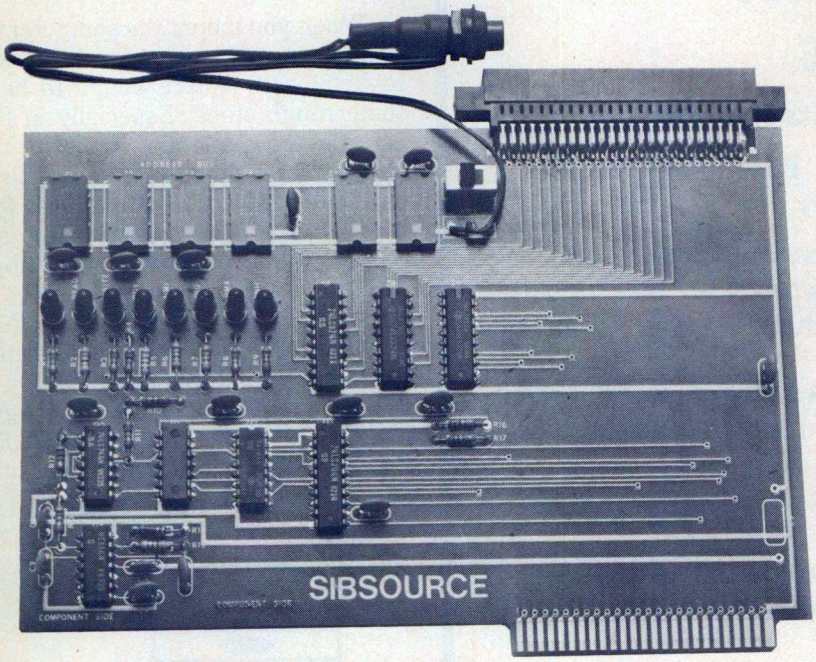
How much does it cost to make a robot without an arm? My robot is a mobile, microprocessor—controlled, electronic entity. It is powered by one 12 volt, 14 amp-hour motorcycle battery. During the summer of '84, I will attach the arm I've been working on. It cost me $160.45 for parts for the robot. But that doesn't include the microcomputer. In 1976 I paid $245 for it. It has a 6502 microprocessor (same as the Apple][). When you start building your robot, if you don't have a computer to dedicate, you will probably choose the Commodore 64, which provides the most hardware for the dollar. It also understands the 6502 instruction set, enabling you to write machine language programs in your Apple][and "download" them to your robot through a simple interface with the C-64. You also get 64K memory with your C64. That's 64 times more than I have in my robot and a third more than I have in my Apple][. One last major item — motors. I was lucky and given two electric window motors. The local wrecking yards are rumored to sell them for 20 dollars apiece — 10 dollars if you do the removal work.

So let's add it up. It all comes to $405.45. When we first started this series, I told you our goal was to build a robot for under $700.00. Looks like I made it. Now, could you? Consider this: $160 for hardware, including wheels, let's say $50 for two motors. $200 for a microcomputer that is compatible with the Apple][. That comes to $410. That's pretty good, and leaves a healthy amount you could spend on books and technical manuals, still spending less than $700 to complete your robot pet.

Here's a partial list of materials. It's partial because the closer I got to completion, the less interested I became in documentation, and the more time I spent being productive. It is complete enough to get you in the range of completion.

- one 12 volt, 14 (or more) amp-hour motorcycle battery
- two 8 inch wheel/tires that can be fixed on a ½" axle
- one 3 to 4 inch caster with rubber wheel and metal frame
- four flanged bushings to support the ½" axles
- thirty to forty feet of ¾" "L" angle aluminum extrusions
- 50-60 10-32 cap head bolts, some 5/8", some 1 inch
- 50—60 washers for above
- 50-60 10-32 hex nuts
- four 12 volt 10 amp relays
- four logic level to 12 volt relay drivers
- 5 74LS05 open collector TTL inverters
- one ACE or PROTO brand breadboard as large as you can get for about $30.00
- three spools of 22 gauge solid wire, each a different color (try Radio Shack)
- one microcomputer
- two 12 volt electric window motors
- some miscellaneous amount of ¼" aluminum to mount the

APPLE BUS MONITOR & EXTENDER



BUS MONITOR – ADDRESS and DATA BUS are displayed simultaneously in Hex. LEDs show READ, WRITE, RESET, IRQ, NMI, RDY, INH and DMA control lines.

CONTROLS COMPUTER SPEED – Allows normal, slow, single step and halt of any programs on any operating system without software changes. Undetectable by software.

A SLOT-INDEPENDENT EXTENDER CARD to help in developing or troubleshooting any Apple compatible cards.

SEE INDIVIDUAL INSTRUCTIONS as any program executes. Valuable as a learning tool or to find that elusive software problem.

ALLOWS YOU TO DO THINGS that you would not be able to do without the use of expensive logic analyzers.

DEALER INQUIRIES INVITED.

SIBSOURCE INC.

15735 Nordhoff St. #52 • Sepulveda, CA 91343
(818) 904-1219

Order by phone or mail.
By mail send check or money order for $225.00.
Calif. residents add 6.5% sales tax.
Add $3.00 shipping.

Circle 23 on Reader Service Card

flanged bushings into
- a drill bit to match the 10–32 screws
- a 10–32 tap
- a 15 by 24 " piece of 1/8" (or so) aluminum plate

Finally, add a lot of dogged determination to complete the project. It really isn't too hard. You just have to set your mind to the task and complete it. Note — I discovered an interesting phenomenon in myself that other experimenters said they shared. I worked out all the problems on the test bench. That is, I connected the computer to the motors outside the robot on the workbench first. I got everything working before installing it, and once I made it work, I knew it would work in the robot. Result: It didn't seem so important to put the stuff inside the robot. I knew what it would do. The challenge was gone. But my friends had heard about the robot for so long that I had to complete it just so they could realize that it would do what I had been telling them. I don't know why they wouldn't take my word for it...

Did you know that you can conduct some experiments of your own, using your Apple][or][+ (I don't know about the *IIe*)? The game paddle port exchanges data with the outside world at TTL logic levels. If that information is meaningless to you, it is *not* time for you to start wiring things up to your Apple, but it is time to read a book like Don Lancaster's *TTL Cookbook.* That should get you started in understanding things electronic.

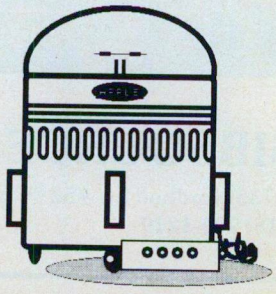
Constructing a robot is a fairly easy process. In a nutshell, you build a framework that will support a battery, computer and motors. Then attach wheels, motors, battery and computer. Connect the electronics and run it. I built a three dimensional rectangular shape using the "L" shaped lengths of aluminum. I'd like to say I used my computer to design the shape, but all I really cared about was the robot's ability to clear doorways. I measured the doorway space, allowed for a margin of sloppy operation, and made a design. The aluminum plate became the base onto which I would attach the frame work. I attached the "L" angles for strength. I used all the same size screws so that any disassembly would require only one or two tools, not a whole raft of different screw drivers or Allen wrenches to adjust some mechanism. I drilled and tapped the holes so each part would be most solidly fastened.

I didn't enjoy the thought of possibly having parts of the superstructure coming loose during normal operation of the robot, so I tapped the holes and lock-washered the nuts. If you have access to some "C" clamps, you can see what the beast will look like before you drill any holes. You can piece it together and clamp the parts into place and "eyeball it" to make visual verification *before* you fasten it. After all your planning and talking, when you start this part, you get the first glimpse and realization that you are actually building a robot, and the project now moves quickly. You won't be able to control yourself. As soon as you see the first vertical piece in place, you will devote every spare moment to completing the robot.

When you mount the computer inside (or on top or wherever), be sure to cushion it. You're bound to have some rough stops. Especially in the early life of your robot. Consider using grommet mounts or foam cushions or some combination that will keep the electronics from being jarred when the rest of the robot stops fast.

Many of you have written in during the course of this project. Thanks again for your words of wisdom. Let me know how *your* projects are going. Write to me in care of A.P.P.L.E. They can usually find me within three to four days.

Good Luck — Now *Get Started!*



Discontinued Model !!!
Strobe M100 Plotter

Originally a $1265 value

NOW AT ONLY $395

Includes: **Model M100 Strobe Plotter**

Configuration Kit - Apple II, II+, IIe, & IBM PC

Enhanced Business Graphics: software for line graphs, pie charts & bar charts.

Strobeview: software for word charts, organization charts & flow diagrams.

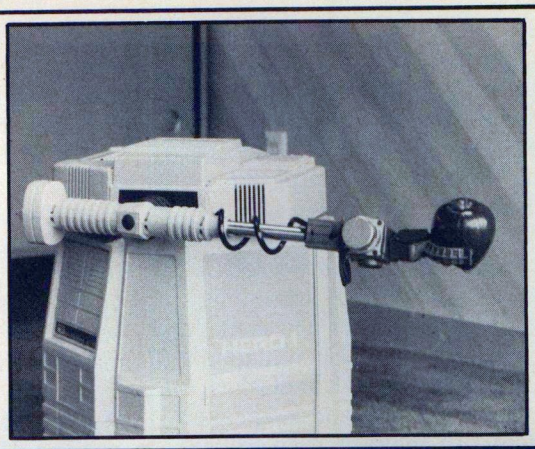
ONLY 150 UNITS LEFT

All purchases - cash, VISA, or Mastercharge
Full 90 day warranty applies.

897-5A Independence, Mt. View, CA 94043
STROBE,INC. (415)969-5130, outside CA (800)451-8108

Circle 32 on Reader Service Card

HEROs and Apples



John Covington and Burt Webb

ROBOTS and computers have always lived in a sort of symbiosis. The first robots were tied to large computers because no small computers were available. Stanford Research Institute's "Shakey" was one of the first mobile platform robots. It was connected to a PDP-10 minicomputer by radio link. With the development of smaller 'minicomputers', the computer could be located on the platform. It still had to be plugged into the wall, which hampered the mobility of the platform.

When the battery operated single board computer (SBC) became available, robot systems could be built that did not require an "umbilical" power cord connecting them to a wall outlet. The SBC could still do as much work as a minicomputer but it did have a number of limitations, such as a small keyboard and an LED readout.

All of today's personal robots are based on SBCs and lack a decent keyboard and screen. This makes it very difficult to enter and modify large programs. Even if a robot has a good keyboard and screen combination, an Apple can be very useful in developing programs. At present, the available robots only have cassettes for mass storage. Anyone who had an Apple][system with cassette storage knows what a pain it could be for loading and saving big programs.

Until bubble memory gets cheaper or disk drives can be battery powered, the personal computer will be a necessary companion for the home robot. To illustrate this relationship, we'll look at a popular personal robot and how an Apple can make it easier to use. A HERO 1 does not require an Apple to operate, but it can be a big help.

HERO 1

The HERO-1 is Heathkit's first try in the personal robot market and one of the first available to the public. The name stands for Heath Educational Robot and that is where it is aimed in the market. It has two personalites; one is a simple home robot and the other is an educational robot in combination with Heath's Robotics and Industrial Electronics course. It is well engineered from a mechanical standpoint, but its massive cable system and lack of a motherboard make it hard to expand.

The robot comes in two forms: assembled and as a kit. The assembled form is more expensive, so many people buy the kit. Like all Heathkit products, the kit is well documented with a lot of helpful hints and designed for easy assembly. It contains 14 separate boards and yards of cables, so it takes some time to construct. Many of the primary cables are already assembled which makes construction a little easier at critical points in the kit building process. You may feel like celebrating when the robot first says "ready" and the HERO-1.0 logo shows up on its LEDs.

Processor

The brain of the HERO is a 6808 microcomputer, which uses the 6800 instruction set. Although no major personal computers use this chip, it is used in Heath's most popular single board computer which is called the ET-3400. The monitor program used in the HERO is a direct steal from the ET-3400 with added features necessary for the special requirements of robot control. This amounts to an extension of the instruction set. The Robot Control Language employs unused 6808 opcodes to cut down on the space requirements for robotic control programs. This means that each instruction uses from one to eight bytes to carry out an action which could be anything from a motor movement to a sensor check. It takes a great deal of machine code to operate a stepper motor because one instruction is required for each tiny step.

Software

No computer of any type can operate without software. In the case of the HERO, the primary software is the monitor and the Robot Control Language. All programs for the robot must be written in either 6800 code or the Robot Control Language or both. The only software that is supplied with the HERO consists of program listings in the four manuals. These are mainly demo programs. A separate demo ROM is available at extra cost. There are a few third party software firms that have created programs for the HERO. Micromation has two such programs. "Storyteller" uses the robot's Votrax to produce self-generated random sentences to tell a story of his adventures. "Poet" uses the same system to produce randomly generated poems which are three line Haiku-style accompanied by a meaningful body movement. At this time, the creation of useful programs is in the hands of the owners. Our forthcoming book on personal robots will have a section on program development for different applications and a list of software vendors.

Communication

Communication with the HERO consists of the Hexadecimal code and function keyboard, the seven segment LED array, and the Votrax

voice synthesizer. The keyboard is a simple key pad on the "head" of the robot below the LED array. It works well enough, but programming at the opcode level with two keystrokes per byte makes even small program entry a chore.

At this time, Micromation has a voice recognition system (VOCREC) available for the HERO. It contains a 65C02 CMOS processor and is made to be interfaced to Micromation's MEMCOM board. VOCREC has a 256 word vocabulary and is speaker dependent. To make it easy to use, VOCREC, a Voice Command Language (VOCOL), has been developed which will let you "talk" your robot thru a training section. The robot will then carry out the commands or store them for later execution. The Votrax speech synthesizer is a phoneme based system which is quite easy to use, although you must work out each word separately. Using the Robot Control Language, you just point to a table of phonemes several times to generate a word. The synthesizer sounds very mechanical but can be understood. The onboard ROM has a number of phrase tables that you can call from your programs so you don't have to construct certain words from the phoneme table. These "ready made" words are also available for use by the system. The primary one is "ready," which the HERO says when it goes back to command mode. If the power falls below a certain voltage the robot will say "low voltage" until you put it on the charger.

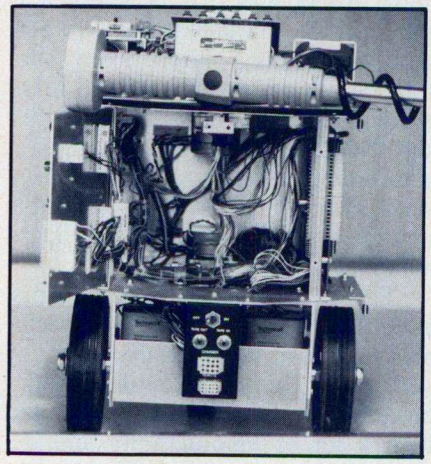
Sensors

The HERO has most of the senses needed by an independent device including sound, light, and motion. It can also tell its distance from things by using its sonar system. The built-in sound sensing system uses the speaker and is not very sensitive. Interface Technology, Inc. came up with a much better two microphone add-on system. The light sensor is just a photocell. Both sound and light are fed to the same small analog/digital converter so the robot can only use one of them at a time.

The sonar system works quite well but can be fooled by other ultrasonic sounds sources. The range of the sonar system is from four inches to about eight feet so it can handle most of the environments in which the the robot will be used. The motion sensor is a simple system based on the same ultrasonic transducers used in the sonar system. It can detect moving objects in an average size room without difficulty. Though some would not consider it a sense, the HERO can also access a real time clock which has hours, minutes, seconds, day, month, and year. This is quite easy to use and is maintained even with the power off.

Manipulation

The HERO arm is a well-built stepper motor manipulator attached to the robot's head. Stepper drive is slow and can only lift one pound which is not very much. Due to power constraints, the motors for any



home robot cannot be very big. The arm can rotate in a sweep of 150 degrees from one side to the other and extend or retract 5 inches. HERO's head rotates 350 degrees, which means the arm can reach things all around the robot. This allows a great deal of flexibility in what it can do. Motor control for all the arm's motors is available from the Robot Control Language. You can also run it through motion patterns with the teaching pendant which is a hand held remote control panel connected by a cable. Once the HERO has been shown a pattern with the pendant, it can then repeat the movements on its own. This system is used on many industrial robots.

Locomotion

The HERO uses its front wheel to get around. A large drive motor controlled by pulse width modulation moves the front wheel forward or backward and pulls or pushes the trailing wheels. Steering is accomplished by turning the drive motor with a steering stepper motor. There is no feedback from the steering motor about its position, so even turning a corner is not easy. The turning circle of the HERO is also quite large because it has to drag the trailing wheels around the turn. The current generation of personal robots, including the HERO, has no braking system. This means that you must reverse the drive motor to slow down and stop or never let the speed get so high that inertia takes over.

Power

Power for the HERO comes from three 6 volt batteries. Two are used to provide 12 volts for both the stepper and the DC motors. The third is used for the logic circuits. A power supply board is used to give a sensing capability for low logic or low drive.

This means that the robot can detect when the batteries are depleted for both systems. When this happens to the logic system, the robot actually says "low voltage" as a reminder to recharge him. A charger is supplied with the robot and it takes about 12 hours to recharge for a run of four hours.

System

All parts of the HERO are tied together quite well; some are even overengineered. The lack of a true motherboard system is a shortcoming, but vendors are creating extensions which solve some of the problems. The HERO is well built and is the only personal robot with a heavy duty arm that can take some abuse.

Conclusions

The primary consideration involved with the purchase of the HERO is whether to buy it as a kit or fully assembled. Since there are few robot repair shops around, it would be best to build the kit if you have the time and skill. It is a *big* kit so, if you have any reservation about building kits, start with a smaller kit and work up to the HERO. Once you have built the Hero, you will also have a simple computer to use for learning hardware and software interfacing. The robot has an input and output port available thru an experimenter's socket: a board into which components can be plugged without

soldering. So far, the HERO is the only personal robot available which has such an experimenter's interface. Overall, the Heath is a solid, well built robot that meets its design goal.

Apple

So you buy or build the robot and you naturally want to program it to do something interesting and/or useful. You will probably start by keying in the demo programs listed in the manuals. When you have to key two strokes per byte you will soon tire of this process and will want to get the good ol' Apple into the act.

Interfacing

This requires a way for the robot and Apple to talk to each other. There are 2 systems that come to mind: Serial and Parallel. The serial system is easier to interface but it is somewhat slower than parallel. If you are a real hardware hacker you get a copy of the HERO's monitor listing and build your own interface. The monitor contains a number of routines that allow the HERO to read and write thru the experimenter's port at baud rates determined by certain bits on the input port. The routines emulate a serial port thru bit 7 of both the input and output ports. Of course, you can use the same ports in parallel mode if you write your own routines.

The next step up is to buy an interface that has all the routines on board. Only one is currently available. It is made by Interface Technology Inc. and is one of the cheapest ways to upgrade the HERO, if you can work under the limitation of the 4K of user RAM supplied with the robot.

If you have bigger plans, more memory is a must. (Remember when you only had 16K RAM on your Apple?) Most of the expansion systems include some way to upload and download programs. One of the first available is from Perobotics. It consists of 8K or 44K of RAM with an RS-232C serial interface. A cassette is available with the transfer routines plus a memory test. Another is available from Interface Technology, Inc., but manufactured by Micromation. It consists of two 8-bit parallel ports plus two 16-bit timers, an RS-232C port, and a ROM chip with the transfer software. It also has a wirewrap area for your own hardware experimentation. It includes 30K of RAM for your programs in 2K increments. This can be replaced by EPROM chips if you want to burn in your own ROM programs.

About the Article

This article is a short review of a readily available robot and its connection to microcomputers in general, with emphasis on the Apple. This review came out of John's building the HERO kit and using the robot. It is not an easy process, and unless you can follow instructions exactly and have a great deal of patience, we would not recommend it for any but the most hardy. You will know the HERO intimately when you are thru, so if something breaks, you can fix it. That is the main reason for building the kit, aside from saving money.

Many of the new robots require you to have an Apple or some type of personal computer. Once you acquire a robot of any type, your Apple will make its programming much easier. Until large LCD screens and small low profile keyboards become readily available, the personal computer will be the primary means of programming personal robots.

About the Book

This article is a sample of what will appear in our forthcoming book on personal robots. It is the result of a great deal of research in the personal robot marketplace and some actual nitty gritty work with the monsters. We hope to make it easier for those who read the book to make an intelligent choice when they go to buy a robot. It is written for consumers, rather than for hardware specialists, because robots should be available to everyone. The book will also provide a wider context with the sections on fictional robots, basic robotic components, robot applications and the future of robots. Robotics will be a new buzzword in the years ahead and, just like the personal computer, the Robot will find its place in our society with help from people like you.

We also welcome any comments by owners of any robots currently available on the open market for possible inclusion in the book as part of the review process. We want the reviews to be even-handed and objective, but we also want to alert users to possible problems with their type of robot.

There are upgrades for the HERO that include new processors and/or new operating systems. Virtual Devices Inc. has a board that has both. The MENOS has a 6801 CPU onboard in place of the normal 6808. They both have the same instruction set and the MENOS board replaces the HERO's original CPU chip in a piggyback arrangment. This allows you to run the original HERO programs without modification. MENOS is an expansion system with RS-232C and 34K RAM plus its own ROM with three operating systems. Also included is software which consists of a high level language and transfer routines. This software comes only in CP/M format so you must have a Z-80 card to run it on the Apple. It allows you to develop software on the Apple and download it without knowing the HERO's internal opcode language. A simulator is also included that allows full development even without a HERO. A duplex 300/1200 baud radio link is also in the works as well as library ROMs. For true expansion, the MENOS has four slots for additional circuit boards. This is the first of a wave of new systems for the HERO and other personal robots.

Code Production

Now you have the robot and a way to communicate with it. If you don't have the Virtual Devices MENOS board, how do you program the HERO from the Apple? An Apple assembler will create opcodes for a 6502. The HERO has a 6808 processor, so you will need crossassembler to produce the opcodes. A number are available from different sources. Micromation Inc. sells the S-C 6800 crossassembler and macros to produce the HERO opcodes. These HERO opcodes are unused by a normal 6800 system, so they have to be created apart from the usual operation of a 6800 assembler. Micromation does require you to buy the S-C 6502 assembler and the cross as an upgrade, so it might be a bit redundant if you have an assembler already. If you have a Softcard, a number of CP/M crossassemblers are available. There is even one available in the public domain, though it doesn't have macros needed to generate the special HERO opcodes. Once you have a

crossassmbler to write your program and RAM space in which to load it, you can get down to some real software development. If a set of HERO macros is not available for your crossassembler, then you must write them. It is not hard to do, because the HERO opcodes are based on the unused opcodes of the 6800 instruction set. Once you have HERO opcode macros, you might want to write macros based on those so that a number of well used routines can be called simply by typing the macros and passing variables.

Languages

The personal robots of today are in the same place that microcomputers were in 1977. They are available but not very useful. Working with the system we have described, someone like you might create the "VisiCalc" of robotics. Of course, the first major programming chore for robots will be the implementation of high level languages. Lisp and Forth will probably lead the pack because they are currently being used in Artifical Intelligence work. There is a great deal of overlap in the problems of Artificial Intelligence and Robotics Control. Forth should be available for all personal robots soon because listings are in the public domain and it requires very little room to implement; a full Forth can be done in 8K. BASIC is slow but also widely available. Robots may give us a good excuse to move away from BASIC as a primary language. Robots need real time sensing and action so the speed of the control language is very important.

Applications

What should your HERO be able to do? With the available sensors, your robot should be able do do quite a bit. Patrolling your house is one possible application. Figuring out the obstacle avoidance algorithm will be the hardest part, but it can be done. Nesting schemes will have to be developed so the robot can find its own charger when the batteries are depleted. Since the HERO can talk, a number of programs are available that can entertain you and your kids. It is easy to add sensors thru the experimenter's port such as a passive infrared sensor to detect living beings. A smoke detector is also easy to add. Like the Apple, the possibilities of a robot are only limited by the capabilities of the robot and your imagination.

Here is a list of suppliers mentioned in the article.

Heath Company
Dept. 025-118
Benton Harbor, Mi. 49022
800-253-0570

Interface Technology, Inc.
Box 745
College Park, Md. 20740
301-490-3608

Perobotics
17072 Emerald Lane #C
Huntington Beach, Ca. 92647
714-847-7846

Micromation, Inc.
9104 Read Brach Rd.
Columbia, Md. 21045
301-730-1237

Virtual Devices, Inc.
P.O. Box 30440
Bethesda, Md. 20814
800-762-ROBO

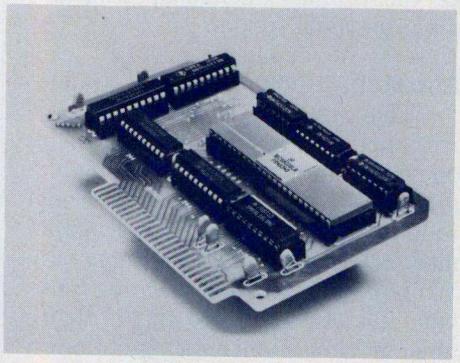
Robotronix, Inc.
Box 1125
Los Alamos, Nm. 87544

About the Authors

John Covington has been in electronics and mechanics since his years in the Navy working on the AN/SPG-49 radar system. His interest in electronics began as a hobby in high school. After the Navy, John worked with microcomputers and bought an Apple in 1978. He began his study of robotics about four years ago and has tried to keep up with the evolution of the field. He also belongs to a local Artificial Intelligence association.

Burt Webb also had electronics as a hobby in High School. He has worked on audio and video equipment, stage lighting equipment, and miniaturized biofeedback devices over the years. His interest in microcomputers began in 1972 when he helped write a book on the coming electronics revolution. He bought an Apple in early 1978 and started Cybercon, his consulting business, in 1980. Burt has written a number of articles and helped in the writing and publication of four books on non-fiction subjects. In his consulting business, he has researched and designed a number of custom interfaces. He is also one of the charter members of A.P.P.L.E., holding membership number 18.

AN AFFORDABLE 68000 System for your Apple!



68000 Mainframe architecture is now available for 8 bit systems with **McMILL**.

McMILL features the new MC68008 member of the M68000 family, and offers Apple II, IIe users a no frills, low cost 68000 based development system.

LEARN MORE ABOUT THE "HEART" OF THE LISA & MACINTOSH SYSTEMS WHILE STAYING WITH YOUR PRESENT II OR IIe SYSTEM.

McMILL IS BOTH SOURCE & OBJECT CODE COMPATIBLE WITH ALL 68000 PROGRAMS, MAKING YOUR APPLE A 68000 BASED DEVELOPMENT SYSTEM!

McMILL is the 68008 based coprocessor board that:
*Fits into any slot on Apple II, IIe or Apple compatible.
*Uses Apple memory and clock for true 6502/68008 coprocessing.
*Comes with ready to run 68000 Fig FORTH and a 6502/68000 MACRO cross Assembler.
*Carrys a full one year hardware warranty.
Features ultimate efficiency in design & power with a fully loaded system for UNDER $300.00.

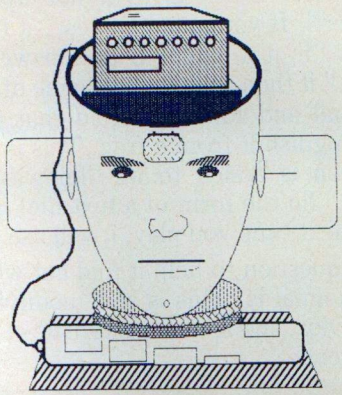
To order, see your local dealer or contact:

STELLATION TWO
The Lobero Bldg.
Box 2342
Santa Barbara, CA 93120
(805) 966-1140 / 966-4964

Lisa/Macintosh, Apple are trademarks of Apple Computer Inc. Fig Forth offered as public domain from Mountainview Press. 6502/68000 Cross Assembler from S-C Software Corp. McMILL, trademark of Stellation Two Inc.

Circle 28 on Reader Service Card

Artificial Intelligence: The Myth and the Reality



Jim Wicker, M.A.

ARTIFICIAL intelligence (AI) is the latest buzzword among the high-tech types. Unfortunately, there is a great deal of misinformation about the subject. One problem lies in coming up with a simple definition of intelligence, much less artificial intelligence.

British psychologist Charles Spearman once defined intelligence as being composed of two factors or "traits." These are a general factor ("g") and a number of specific factors relating to particular activities. Specific factors might be things like mathematical aptitude or verbal skills, while the g factor would show up in things like abstract relationship tests.

In the U.S., Spearman's theory was modified to what is called a multiple-factor mode. According to this theory, there are several broad group factors, each of which has a different weight in different I.Q. tests. If, for example, you have an I.Q. test that emphasizes vocabulary, it will mainly measure the verbal factor of intelligence, while another I.Q. test that emphasizes arithmetic would basically measure the computational factor of intelligence.

Unfortunately, no one is really sure which of these theories (if either) really describes intelligence. In fact, some psychologists just say that intelligence is that quality measured by I.Q. tests, and let it go at that.

One pragmatic way to define artificial intelligence is to perform what is known as the Turing test. Alan Turing proposed that if a human observer were to communicate via a data link (a teletype or CRT/keyboard) with either another human being or a computer, and could not determine whether it was a computer or a human being, then that computer could be said to possess artificial intelligence.

This sounds good, but can be very tricky. I remember one afternoon when I was communicating over my Apple via a modem with a friend of mine who also owns an Apple. My son (age 15 at that time) came in, and I suggested he try out the new "artificial intelligence algorithm" that I was working on. He was instructed to type whatever he wanted into the Apple. After a couple of minutes, he said "No way, Dad. What are you up to?" He understands the limitations of computers, and particularly of microcomputers, far better than some of the "experts" in this field.

Using Turing's approach, for example, it would be very difficult to determine whether you were playing a human chess player or a machine. The new chess algorithms are very good, better than most players, in fact. But, is this artificial intelligence?

The trouble here is that the chess program achieves the same end result as the human player, but by very different means. In playing chess, very few human players look more than a few moves ahead, and none of them exhaustively consider all possible ramifications of each move. They move based on a kind of intuition, a feeling of what "should be" the best move, not an analysis of all possible moves. The chess program, on the other hand, does an analysis of all possible moves down to some level, and then makes its move based on this information, supplemented perhaps by a "library" of good responses.

The chess program is unable to use an intuitive approach, and yet even animals can do this. There is a classical study in psychology, where a bunch of bananas are hung out of reach of a chimpanzee. Near the chimp are two sticks, neither one long enough to reach the fruit. It is, however, possible to join the sticks together. Very often the chimp does just this, fastening the sticks together and knocking down his dinner, thus demonstrating the intuitive aspect of intelligence.

So far, I have seen *no* artificial intelligence algorithms that will do this. However, this is not to say that there is no such thing as artificial intelligence. There is — it is just that it relates only peripherally to human or animal intelligence.

As far as I am concerned, AI is a class of algorithms for computers that fall into one or more of several categories: expert systems, adaptive algorithms, and choice pruning algorithms. I'll try to explain what each of these is.

Expert systems are an attempt to have the computer serve as an expert resource for some activity. That is, the computer will have available the expert knowledge needed for some particular process. If, for example, you had an airplane that was equipped with an expert system in navigation, then the pilot can query the system anytime he needs navigation data or navigation techniques. The expert system would know all about radio beacons, compass headings, winds, and the like.

Adaptive algorithms are programs for computers that allow them to modify their own responses based on prior or ongoing "experience." A good example of this is the game "Animals" that is on the DOS 3.3 Master disk. In this game, the Apple tries to name an animal that you are thinking about. It does this by attaching conditions to the name. For example, as the game begins, the

computer asks "Does it live in the water?" If you answer "yes," it then asks "Is it a frog?" If the answer is "no," it then asks for the name of the animal and for a question that will distinguish it from a frog. This information is added to its "knowledge base" (in the form of a text file), and the next time you play, it will use the new question to help it find out what the animal is. This is self-modifying behavior, and is a characteristic of intelligence.

Choice pruning algorithms are techniques for making decisions. That is, these are heuristic algorithms. This technique is used in chess programs to help decide what move to make next. Instead of considering all branches of all decision points, these algorithms limit the choices, based on available data. In a chess game, for example, if both bishops are gone, the program need not consider those classes of moves that involve bishops. This is, in reality, just a way of reducing the computational load required and is used in conjunction with expert systems and daptive algorithms.

What is AI, then? Well, it is not human intelligence, converted into machine format. Human intelligence involves a great many factors that computers simply cannot emulate, including intuitive behavior, self-awareness, and non-verbal processing. HAL is not here yet, and I don't think it ever will be. However, systems that can respond in useful and effective ways to human (that is, fuzzy) inputs are here and are certain to become an ever larger part of our culture. This includes things like speech recognition, systems that "tailor" themselves to individual inputs, and perhaps even mobile, self-adaptive robots. But, human they aren't.

For those of you who are interested in AI for the Apple, a very good book is: *Projects in Machine Intelligence for Your Home Computer* by David L. Heiserman, TAB Books, Inc. #1391, Blue Ridge Summit, PA 17214. The programs are fun, as long as you don't take them too seriously.

Using an

Susan K. Lynch, RB Robot Corp.

PERSONAL computers in general, and Apple computers in particular, are a common sight in today's schools, homes, and businesses.

Personal, intelligent robots, though not quite so common now, may well be just as popular as the personal computer in the very near future.

People expect personal computers to be easy to understand and use. Well-written and carefully documented software has provided that ease for many applications. Robots are also expected to be friendly and easy to communicate with, even more so than personal computers.

* THE PORTFOLIO MINDER SYSTEM *

Softcell, Inc. would like to introduce you to the Portfolio Minder System, the first truly easy to use portfolio tracking system. Within minutes you can have the program running and managing your portfolios. All you need is an Apple Computer with 48K, two disk drives, and a 132 column printer.

Simplicity
We at Softcell, Inc. believe that computers should do all the work for you. Therefore, we have written the Portfolio Minder System for people who don't like spending three weeks learning how to use a program. In fact, it's so simple to work with, that you can have the system up and running in a half-hour.

Operating Speed
The Portfolio Minder System is designed to take as little of your time as possible during operation. Unlike other portfolio tracking programs, the Portfolio Minder allows you to enter transactions from all accounts at one time. It automatically files them in the correct portfolios for you.

Dow Jones Quote Service
Incorporated in the Portfolio Minder System is the ability to update all current prices of stocks, bonds, options, mutual funds, and U.S. Treasury notes using the Dow Jones Quote Service. All you need is a Hayes Micromodem II and a Dow Jones password.

Printouts
The Portfolio Minder System provides readable, well organized printouts. The program calculates an account's status, unrealized gains(losses), realized gains(losses), yearly income, and percentage change of a given security. In addition, a cross reference report lists all securities with their corresponding total holdings broken down by account name and number.

Extra Options
In addition to the capabilities listed above, the program allows editing by account or on a global basis, and manual inputting of current prices.

The Portfolio Minder System is available for $125 plus $1.50 for shipping.

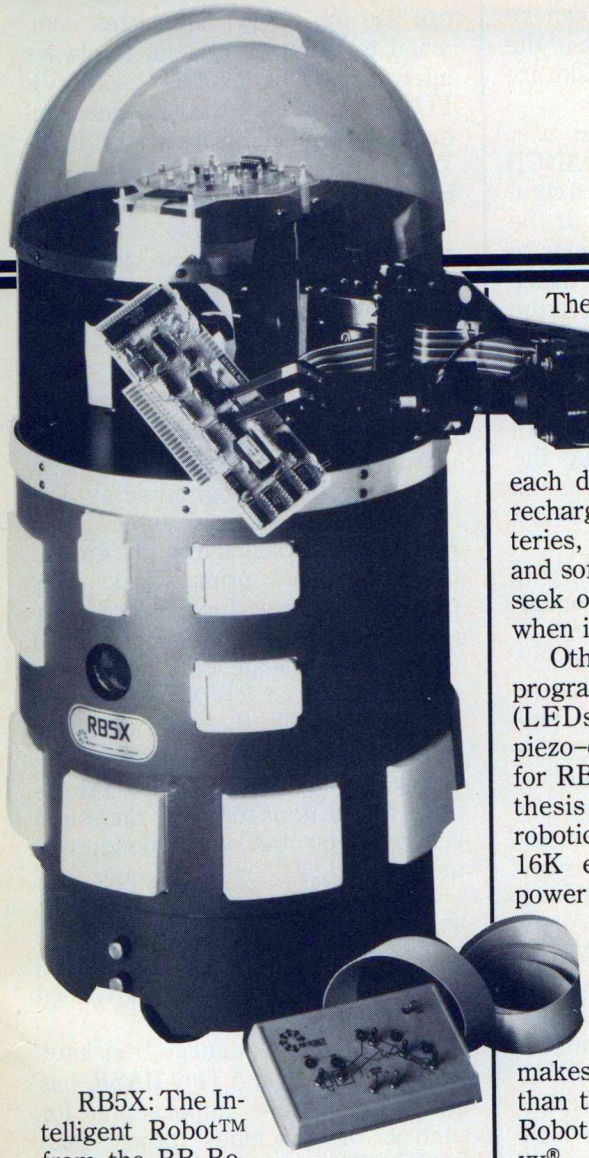
SOFTCELL, INC.

P.O. Box 4031
Greenwich, Ct. 06830

Apple is a registered trademark of Apple Computer, Inc.; Hayes is a registered trademark of Hayes Microcomputer.

Circle 24 on Reader Service Card

Apple to Speak English to a Robot



RB5X: The Intelligent Robot™ from the RB Robot Corporation in Golden, Colorado, is the world's first mass-produced, programmable robot designed especially for home use, educational purposes, and experimentation. RB5X™ has an on-board microprocessor with 8K user- available read-write memory (RAM); the robot's microprocessor can be programmed in Tiny BASIC using any computer with an RS-232 serial communications interface. RB5X is cylindrical in shape with a smoked polycarbonate dome; stands a little under two feet tall, weighs 24 pounds, travels on synthetic rubber drive wheels and casters, and navigates using a sonar sensing system, eight tactile sensors, and an infrared photodiode located under its carriage.

The robot travels forward or backward at about 4" per second, rotates around its center point, and pivots forward and backward on each drive wheel. It is powered by rechargeable, sealed lead-acid batteries, and comes with a recharger and software routines that allow it to seek out its charger and plug itself when its batteries are low.

Other standard features include programmable light-emitting diodes (LEDs), and a programmable piezo-electric horn. Current options for RB5X include a voice/sound synthesis package, a sophisticated robotic arm with five axis of motion, 16K extended memory (RAM), a power pack for an additional 10 amp-hours of battery power, and a series of preprogrammed application software modules.

But the thing that really makes RB5X unique, and friendlier than the other robots in its class, is Robot Control Language™ with Savvy® — or RCL™ for short.

Imagine using your Apple computer to speak English to a robot. It does sound a bit strange, doesn't it? RCL is a programming language software system available for use with the Apple *IIe* and Apple][+ computers. RCL requires only the use of simple English phrases, such as "Move forward," to control RB5X. RCL allows the robot user to bypass the Tiny BASIC programming process the robot usually requires, and is, of course, far friendlier than most of the programming languages required by other robots.

RCL combines hardware and software to form a complete robotics software development system for RB5X. This system is both easy enough for non-programmers to use and complex enough to challenge more experienced programmers.

RCL comes with a "primary task" vocabulary. There are five different types of primary tasks provided in the RCL vocabulary: action commands, programming commands, parameter-passing functions, error - checking and compiler routines, and sample robot tasks. To program RB5X, users select various primary tasks and construct a "robot task" using their Apple computers. Action commands instruct RB5X to perform a specific activity, for example, MOVE FORWARD or PIVOT ON LEFT CLOCKWISE. Programming commands are used for initialization, branching, looping, reporting errors, and other programming conventions. To pass specific information to the robot about how far to move or how long to perform a task, parameter-passing functions are used, such as MOVE TIMED FORWARD for this many seconds: <1>. Parameter-passing functions are also used as programming conventions; for example, DEFINE A VARIABLE called <1>.

Error-checking and compiler routines ensure the proper functioning of RCL primary tasks, and provide functions necessary for all the primary tasks such as incrementing statement numbers and counters, and inserting necessary program delays. The sample robot tasks are examples of robot routines that use the primary tasks. Users can program their robots to perform the sample tasks or use the sample tasks as part of other robot tasks. Skillful use of these various primary task types can result in a high level of complexity within a single robot task.

A robot task is the RCL equivalent of a program for RB5X's microprocessor. For example, the 12-line robot task in Figure 1 instructs the

robot to wait for a bumper press, then to move forward and flash its lights until it makes contact with an obstacle with one of its bumpers, and finally to stop moving and turn off its lights.

Figure 1.

```
1. Does PREPARE THE ROBOT
2. and BEGIN A LOOP
3. and EXIT IF ANY BUMPER
   TOUCHED
4. and REPEAT THIS LOOP
5. and TURN ON THE FLASHING
   LIGHTS
6. and MOVE FORWARD
7. and BEGIN A LOOP
8. and EXIT IF ANY BUMPER
   TOUCHED
9. and REPEAT THIS LOOP
10. and STOP ALL MOTION
11. and TURN OFF THE FLASHING
    LIGHTS
12. and END
```

Each of the lines in Figure 1 is a primary task from the RCL vocabulary; the primary tasks are straightforward and easy to understand. By contrast, look at Figure 2, the Tiny BASIC equivalent of Figure 1.

Figure 2

```
10 @#7803 = #98
20 REM START A LOOP
30 Y = @#7800
40 IF Y<255 GOTO 70
50 GOTO 30
60 REM EXIT TO HERE
70 X = #40
80 GOSUB 3100
90 @#7802 = #09
100 REM START A LOOP
110 Y = @#7800
120 IF Y<255 GOTO 150
130 GOTO 110
140 REM EXIT TO HERE
150 @#7802 = 0
160 X = #BFTR
170 GOSUB 3120
300 0 GOTO 3300
310 0 U = @#7801
310 5 U = U OR X
311 0 @#7801 = U
311 5 RETURN
312 0 U = A#7801
312 5 U = U AND X
313 0 @#7801 = U
313 5 RETURN
330 0 REM END OF PROGRAM
```

While the user is constructing the robot task, the task is written on to the diskette containing the RCL primary task vocabulary. To have RB5X execute the robot task, the user instructs RCL to act as a foreign language interpreter and translate the RCL commands into the Tiny

ticularly valuable for programming RB5X is the extensibility it offers RCL. Because each robot task is, like the RCL primary tasks, also an actual Savvy task, the robot tasks can be used to build a new robot task in the same way the primary RCL tasks are used.

For example, suppose the user writes a robot task called DANCE that instructs RB5X to execute a complicated series of movements. If the user wants RB5X to invite guests to dance at a party and then to dance with them, he or she can insert the robot task DANCE into the new robot task, INVITE AND DANCE, simply by typing the title as a robot task line. It is unnecessary to tell RCL to find, fetch, and run DANCE; typing the name implies all those operations within the new robot task, and Savvy does them automatically.

Another useful Savvy feature is adaptive pattern recognition processing (APRP). APRP allows Savvy to figure out the meaning of a command that does not exactly match any entry in its vocabulary, and to act according to that meaning. This is a marked difference from most programming languages, where an inaccurately typed command usually elicits an error message. Savvy tolerates misspellings, and interacts with the user to interpret a previously unrecognized word or phrase. For example, if you enter the command, GO FORWARD, Savvy searches its memory, including the RCL primary task vocabulary, presents you with as many phrases as it has that resemble GO FORWARD, and asks you which one you want.

I'm not sure what GO FORWARD refers to. Is it one of these:

```
1. GO CLOCKWISE
2. GO COUNTERCLOCKWISE
3. MOVE FORWARD
4. or None of the above?
```

Please type the number of the correct answer:

Assuming you really meant MOVE FORWARD, which is a command from the RCL vocabulary, you would type 3 and press the return key. If you had typed MOVE FOREWARD, or MVOE FORWARD, rather than GO FORWARD, Savvy's APRP system is smart enough to figure out that you really mean MOVE FORWARD, and will execute that command without presenting you with alternatives.

If you happen to prefer GO FORWARD to MOVE FORWARD, you can use the Savvy "associate" command to teach the system to always interpret GO FORWARD as MOVE FORWARD. Using the associate command does not eliminate MOVE FORWARD from the RCL vocabulary, however, for Savvy always recognizes new commands in terms of the original commands it "learned" at the factory. Savvy allows RCL to be a dynamic system; the user trains it to respond to his or her communication style.

The advantages of using RCL are obvious for the beginning robot programmer, but why would it interest more advanced programmers? The answer lies in RCL's extensibility. RCL can be extended from the ground up by adding new primary tasks. This provides challenge for advanced programmers because it requires a good working knowledge of Tiny BASIC programming, of RB5X, and of Savvy. For example, there are currently no RCL primary tasks for simultaneously moving several joints of the optional RB arm, or for using the sound generator on the voice/sound synthesis card.

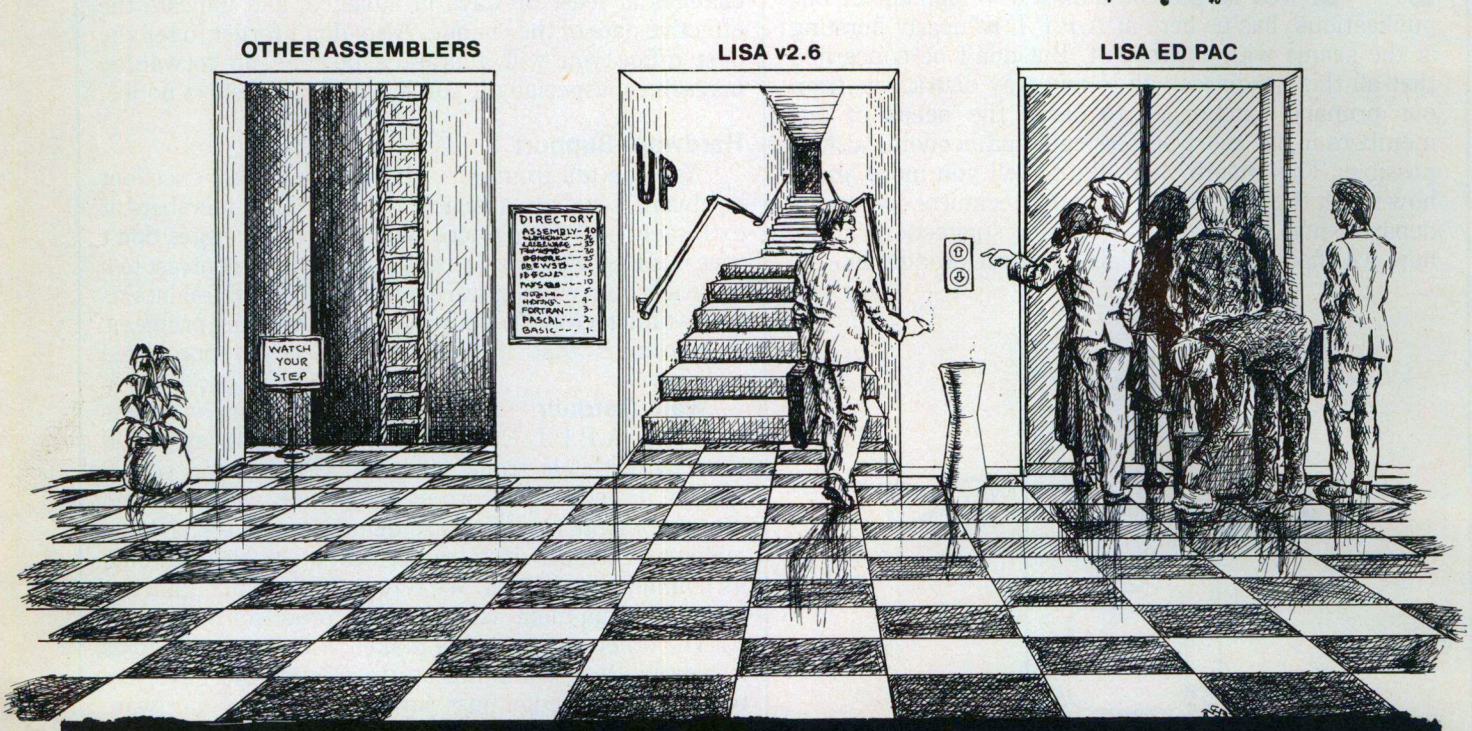
Learning enough about Savvy, Tiny BASIC, and RB5X to facilitate developing new primary tasks is also, in itself, a challenge for advanced programmers.

RCL's role as language translator between English and Tiny BASIC has interesting future implications for control of machines other than RB5X. Theoretically, RCL could work from the same English commands to compile programs in many languages; Pascal COBOL, assembly language, etc.; whichever language the machine to be controlled requires.

In an era of increasing automatization and computerization, it is important for people to comfortably control the machines they encounter at home, on the job, and at school. RCL is a major breakthrough in the development of facile communication between humans and machines because it is *friendly,* does not demand a great deal of technical or computer programming experience to use, and is sophisticated enough to be extended and tailored for specific applications.

There are three ways to learn 6502 Assembly Language on your Apple Computer:

Hard Easy Easiest



Introducing the Easiest Way: The LISA Ed Pac™

You can't deny that learning assembly is extremely important for you if you want to make the most of your work. If assembly language wasn't so important, why are almost all of the top selling programs available for the Apple II written in assembly language? But let's face it, learning 6502 assembly language isn't a piece of cake. At least not until now. Because now there's the LISA Education Package™ from Lazerware. It'll have you up to speed with assembly language in a fraction of the time it would otherwise take.

The LISA Ed Pac™ begins with LISA v2.6, the favorite assembler of beginners and professionals alike. More Apple owners have learned 6502 assembly language using LISA than all the other assemblers combined. More tutorial material is available for LISA, including books by D. Fudge, R. Hyde, W. Maurer, and R. Mottola. Randy Hyde's 300-page *Using 6502 Assembly Language* is included in the LISA Ed Pac™.

Next we threw in SPEED/ASM™, a set of 6502 subroutines that make programing in assembly language as easy as BASIC. And for those who want to see how it's done, the SPEED/ASM source listings are also included. We also included the LUD #1 (Lisa Utility Disk #1) which includes an extended editor for LISA and a LISA source file listing utility. Finally, we added MAXWELL'S Debugger™ to the LISA Ed Pac. This ultra-powerful debugger/monitor makes learning and debugging 6502 assembly language a breeze.

LISA Ed Pac Price $149.95, A $229.75 Value (suggested retail).
Available at dealers everywhere, or directly from:

LAZERWARE

For a copy of Lazwerware's *A Guide to Purchasing a 6502 Assembler for Your Apple II or Apple //e*, write us at Lazerware, 925 Lorna St., Corona, CA 91720 or call us at (714) 735-1041
Note: LISA, LISA v2.6, LISA Ed Pac, LISA Educational Package, Speed/ASM, Maxwell's Demon, and LUD are trademarks of Lazwerware
Apple, Apple II, and Apple //e are trademarks of Apple Computer, Inc.

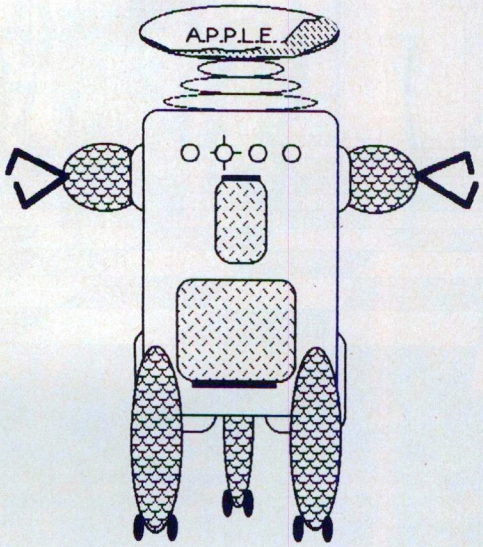
Circle 14 on Reader Service Card

MAY 1984

MEMBER BILLBOARD

Don Elman

As you can surely sense from even a quick perusal of this issue, the anticipatory thrill generated by the new Apple products, as well as the continuing development of our publications, has us here at A.P.P.L.E. nearly bursting at the seams with excitement. But don't be concerned that all this headiness will in any way distract us from our primary function — to serve the needs of our members in the most professional and economical way possible. This month we'd like to tell you more about how to get the best service from our technical operations departments, and then ask for your suggestions about how we should support Apple's newest product line.



Missing Issues

From your gratifying feedback, we are well aware of how important the timely receipt of each *Call -A.P.P.L.E.* issue is. Last month we addressed some possible remedies for the uncertain delivery speed of the USTS (United States Turtle Service). When several days have passed beyond the usual arrival date of an issue, you may rightly become concerned that perhaps it has become lost along the way. Rarely does this happen, but when it does we are more than happy to send you a replacement.

Because the mailing time is highly variable and beyond our control, you can help our operations staff do their job more efficiently by keeping in mind the following policy before inquiring about a missing issue: *If you have not received one month's magazine by the time the next month's issue arrives, we will replace the missing issue without charge if all address information on file is current and correct.* Most often these problems arise when people move, so it is important that you notify us of address changes at least 30 days in advance, and indicate the effective date of the change. Also, don't forget to tell the post office you will guarantee postage on forwarded magazines, especially if you leave town on short notice.

Hardware Support

You can tell from our catalogs how many excellent hardware items are available to A.P.P.L.E. members at extremely attractive prices, but what those pages don't reveal is how we back up these products. In contrast to a mere discount catalog warehouse, A.P.P.L.E. maintains a full-time technical department to handle any problems or questions that may arise with newly-purchased equipment.

Walt Gardner, our hardware technician who has been with A.P.P.L.E. for about 1 year, has always loved working with electronic gadgets. From grade school science projects to a current fascination with hi-fi and digital components, Walt has demonstrated a knack for making electronic circuitry perform at, or even beyond, its nominal capabilities. We rely on Walt to maintain our in-house equipment as well as the products we sell.

The other attribute that makes Walt invaluable to us is his ability to communicate to others who have less technical knowledge just how to get their hardware operating properly. If you ever need to call us with a problem about hardware, chances are good that the clear, friendly voice you'll hear will be Walt's. In fact, he plays an integral role in our 15-day return policy. The reason you are required to call us for authorization before returning a product isn't just a matter of bureaucratic red tape. Rather it is to give you a chance to discuss the problem with a sympathetic and knowledgeable technician before giving up on the product. More often than not, a solution can be found by phone — saving all of us the frustration and expense of an unnecessary return.

Keeping Up with the *IIc*

So much will be said about the Apple *IIc*, both elsewhere in this issue and in other forums, that we won't dwell on it here — except to remind you that, thanks to Apple's wise decision to retain compatibility between the *IIe* and the *IIc*, the vast majority of products currently available through A.P.P.L.E. to our members will work with the new machine. To assist us in serving you as this new computer gets off the ground, we invite you to let us know what additional products and information — particularly those aimed at *IIc* users — you would like to see us provide.

A Graphics Tool Kit

(Part 2)

Randi J. Rost

HELLO again! Welcome to Part 2 of The Graphics Toolkit. You will recall that previously we discussed some of the basics of Apple graphics and laid the groundwork for building our collection of machine language subroutines. We developed a lookup table, called the YTABLE, that will significantly speed up a lot of our graphics routines. We also came up with four routines (INIT, HGR, HGR2, and BKGND) to start off our toolkit. This month we'll look at three more routines: HCOLOR, HPLOT and HPOSN and write a short program to test these new routines.

State Elements

Your Apple is a finite state machine, that is, a machine that can only be in one of a finite number of states at a given instant. You can think of the Apple as containing a very long string of binary digits (bits). Whenever any bit is changed, the state of the machine changes. Each bit is then a "state element" — an atomic element of the machine, that, when changed, implies the state of the machine is changed as well.

What we are dealing with in this series is a subset of the Apple's memory space that looks like a simple graphics machine. State elements in this graphics machine include such things as the current color and the current position. When either of these change, the graphics machine is in a different state.

Two of the routines we'll talk about this month are responsible for modifying state elements in the graphics machine. HCOLOR, as you may have guessed, will allow us to change the current plotting color and HPOSN will let us change the current plotting position. The third routine, HPLOT, modifies two state elements. First, it calls HPOSN to set a new current position, then it modifies the pixel at that position. Each pixel in the memory of our graphics machine is also a state element.

HCOLOR, as you can see, is a very simple routine. It is passed a value from 0–7 in the A register, transfers that value to the Y register, then does an indirect indexed load, using the pointer to the color table. The color table (part of YTABLE) contains just eight bytes which are color masking bytes for each of the eight high-res colors. This value is then stored in the location COLOR and will be used for all subsequent point and line drawing operations. (It should be noted here that in the interests of efficiency, very little checking is performed on the values passed to these graphics routines. If you pass HCOLOR a value of 147, expect to get something weird when you draw on the screen!)

Let's take a look at HPOSN next. The purpose of HPOSN is to change the current position to the position of the point passed in the registers. This can be used to position one endpoint of a line or plot a single point. What do we need to do to calculate the byte address given a high-res screen coordinate? First we need to get the base address of the row. This is done using a table lookup as explained in the last column. The low byte of the base address is obtained directly from the table using the y-coordinate as an index. The high byte is obtained similarly, then OR'd with the current value of HGRPG. The resultant two-byte address is the address of the first byte of the row we want. This address is stored in TEMPPTR.

We can now use the XDIVTBL and XMODTBL values from the table in order to calculate the precise byte position and bit position within that byte. One complication arises since the two X tables are greater than 256 bytes long. Each of the X table pointers is incremented by one if the x-coordinate is greater than 256. Later on, we'll need to decrement these pointers if they are incremented here. The XDIVTBL pointer gets us the value of x/7 so that we can get a byte offset to calculate the precise address needed. Since there are 40 bytes of memory per high-res screen row, we'll get a value from 0–39. This byte offset value is then stored in BYTENUM.

The XMODTBL pointer is also indexed by the x-coordinate to obtain the value of x mod 7 + 1. We now subtract one to get a value in the range 0–6 and use that as an index into the pixel position table. This gives us a mask for the position of the bit within the byte. This masking value is stored in POSBIT. Now, with the values of TEMPPTR, BYTENUM, and POSBIT, we know the precise location in memory of the high-res coordinate specified.

There is one more operation that HPOSN performs before returning. Remember our discussion last time about how the bit pattern needed to be shifted in order to maintain a constant on-off-on-off pattern from one byte to the next? HPOSN checks to see if BYTENUM is odd or even. If it is odd, the current color mask byte is shifted so that we get the right color. Whether it's shifted or not, the correct color mask value is left in SHFTCHK.

HPLOT now has it easy. The only trick is to plot the new point in the desired screen memory byte without affecting the other six pixels in that byte. HPLOT calls HPOSN, passing

Listing 1

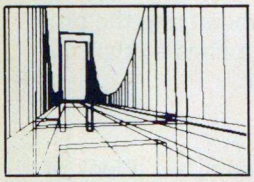
```
                        1               ORG   $8BC
                        2
                        3    SHFTCHK    =     $1C
                        4    TEMPPTR    =     $26
                        5    POSBIT     =     $30
                        6    POSTABL    =     $4A
                        7    XDIVTBL    =     $4C
                        8    XMODTBL    =     $D0
                        9    OLDXLO     =     $E0
                       10    OLDXHI     =     $E1
                       11    OLDY       =     $E2
                       12    COLOR      =     $E4
                       13    BYTENUM    =     $E5
                       14    HGRPG      =     $E6
                       15    YTABL      =     $EC
                       16    YTABH      =     $EE
                       17    COLTABL    =     $FE
                       18    SHFTCOL    =     $8B0
                       19
                       20    PIXPOS     DS    1
                       21    *
                       22    * The HCOLOR routine will change the
                       23    *  current plotting color to the color (0-7)
                       24    *  that is passed in the A register
                       25    *
08BD: A8               26    HCOLOR     TAY
08BE: B1 FE            27               LDA   (COLTABL),Y ;Load appropriate color mask byte
08C0: 85 E4            28               STA   COLOR       ;Save it for future use
08C2: 60               29               RTS
                       30    *
                       31    * Call HPLOT with a point to be plotted.
                       32    * The high byte of the X-coordinate should
                       33    *  be passed in the A register, X-LO in the
                       34    *  X register, and the Y-coordinate in the
                       35    *  Y register.
                       36    *
08C3: 20 D3 08         37    HPLOT      JSR   HPOSN       ;Find pixel adrs
08C6: A5 1C            38               LDA   SHFTCHK     ;and correct color mask
08C8: A4 E5            39               LDY   BYTENUM
08CA: 51 26            40               EOR   (TEMPPTR),Y
08CC: 25 30            41               AND   POSBIT      ;Turn on correct point
08CE: 51 26            42               EOR   (TEMPPTR),Y
08D0: 91 26            43               STA   (TEMPPTR),Y ;Store byte back on screen
08D2: 60               44               RTS
                       45    *
                       46    * Call HPOSN with a point to position but
                       47    *  not plotted.  This point should be passed
                       48    *  in the same manner as HPLOT (described above).
                       49    *
08D3: 85 E1            50    HPOSN      STA   OLDXHI      ;Save X-HI
08D5: 86 E0            51               STX   OLDXLO      ;Save X-LO
08D7: 84 E2            52               STY   OLDY        ;Save Y
08D9: B1 EC            53               LDA   (YTABL),Y
08DB: 85 26            54               STA   TEMPPTR     ;Find Y base adrs
08DD: B1 EE            55               LDA   (YTABH),Y
08DF: 05 E6            56               ORA   HGRPG       ;Get correct HGRpg
08E1: 85 27            57               STA   TEMPPTR+1   ;Store adrs in TEMPPTR
08E3: A4 E1            58               LDY   OLDXHI
08E5: F0 04            59               BEQ   LT256B
08E7: E6 4D            60               INC   XDIVTBL+1   ;Increment XTABLE ptrs by
08E9: E6 D1            61               INC   XMODTBL+1   ;one page if X>255
08EB: A4 E0            62    LT256B     LDY   OLDXLO
08ED: B1 4C            63               LDA   (XDIVTBL),Y
08EF: 85 E5            64               STA   BYTENUM     ;Find byte position in row
08F1: A4 E0            65               LDY   OLDXLO
08F3: B1 D0            66               LDA   (XMODTBL),Y
08F5: A8               67               TAY
08F6: 88               68               DEY
08F7: B1 4A            69               LDA   (POSTABL),Y
08F9: 85 30            70               STA   POSBIT      ;Find pixel position in byte
08FB: 8C BC 08         71               STY   PIXPOS
08FE: A5 E1            72               LDA   OLDXHI
0900: F0 04            73               BEQ   LT256D
0902: C6 4D            74               DEC   XDIVTBL+1   ;Restore XTABLE ptrs if
0904: C6 D1            75               DEC   XMODTBL+1   ;X was > 255
0906: A5 E5            76    LT256D     LDA   BYTENUM
0908: 18               77               CLC
0909: 4A               78               LSR
090A: A5 E4            79               LDA   COLOR       ;Shift color masking
090C: 85 1C            80               STA   SHFTCHK     ;byte if necessary
090E: B0 A0            81               BCS   SHFTCOL
0910: 60               82               RTS

--End assembly, 85 bytes, Errors: 0
```

Listing 2

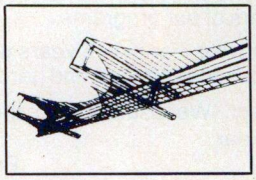
```
                        1    *
                        2    * Title: Demo Program 2
                        3    * Function: Test HCOLOR, HPLOT and HPOSN routines
                        4    *
                        5            ORG   $6000
                        6
                        7    INIT    =     $803
                        8    HGR2    =     $85E
                        9    HGR     =     $865
                        10   BKGND   =     $877
                        11   HCOLOR  =     $8BD
                        12   HPLOT   =     $8C3
                        13   HPOSN   =     $8D3
                        14
6000: A9 1B             15   DEMO2   LDA   #$1B     ;Load A with YTABLE adr hi
6002: A2 00             16           LDX   #$00     ;Load X with YTABLE adr lo
6004: 20 03 08          17           JSR   INIT     ;Set up lookup table ptrs
6007: 20 65 08          18           JSR   HGR      ;Init hi-res graphics on page 1
600A: A9 05             19           LDA   #05      ;Change plotting color to orange
600C: 20 BD 08          20           JSR   HCOLOR
600F: A9 BF             21           LDA   #191     ;Init (X,Y) to (191,191)
6011: 8D 48 60          22           STA   TEMPX
6014: 8D 49 60          23           STA   TEMPY
6017: A9 00             24   LOOP1   LDA   #0
6019: AE 48 60          25           LDX   TEMPX
601C: AC 49 60          26           LDY   TEMPY    ;Draw line from (191,191) to (0,0)
601F: 20 C3 08          27           JSR   HPLOT
6022: CE 48 60          28           DEC   TEMPX
6025: CE 49 60          29           DEC   TEMPY
6028: D0 ED             30           BNE   LOOP1
602A: A9 06             31           LDA   #06      ;Change plotting color to blue
602C: 20 BD 08          32           JSR   HCOLOR
602F: A9 BF             33           LDA   #191
6031: 8D 48 60          34           STA   TEMPX
6034: A9 00             35   LOOP2   LDA   #0
6036: AE 48 60          36           LDX   TEMPX
6039: AC 49 60          37           LDY   TEMPY
603C: 20 C3 08          38           JSR   HPLOT    ;Draw line from (191,0) to (0,191)
603F: EE 49 60          39           INC   TEMPY
6042: CE 48 60          40           DEC   TEMPX
6045: D0 ED             41           BNE   LOOP2
6047: 60                42           RTS
                        43   TEMPX   DS    1
                        44   TEMPY   DS    1

-End assembly, 74 bytes, Errors: 0
```


Don't Settle For One Point of View

Step into the world of 3D



Create, edit, move, rotate, animate . . . produce amazing graphics effects.



It's easy - with Metalogic. A comprehensive interface to SubLogic's powerful graphics package. But forget the hex, PEEKs and POKEs. Just combine Metalogic's "&" commands with regular BASIC statements and you've mastered the 3rd dimension.

Why settle? Let your possibilities be limited only by your imagination — not your software.

Req. SubLogic A2-3D2; Apple][Plus & RAM Card (or Apple //e); DOS 3.3
$75 (or $180 including complete SubLogic graphics package)

Apple][Plus, Apple //e, DOS 3.3 are trademarks of Apple Computer Corp. A2-3D2 is a trademark of SubLogic Corp. Metalogic is a trademark of Artemis Systems, Inc.

ARTEMIS SYSTEMS, INC.
41 Parkview Drive, Millburn, N.J. 07041
(201) 564-9333

MasterCard VISA

New Jersey res. add 6%
3rd class postage pd all orders
add $2 for 1st class mail.

Also from Artemis

Telephile **$20**
A database to keep tabs on all your important contacts-friends, relatives, business associates
- Simple to use
- Complete screen editing
- Flexible search includes wildcards
- Hard copy in variable formats
- Approx. 200 116-character records in each file
- Includes Name, Address, Phone, & Comments Fields

Pascat **$25**
Catalog all your Pascal disks for easy reference
- Automatically reads and sorts all files
- Creates master catalog and saves it
- Screen list or hardcopy
- Flexible search capabilities-including wildcards
- Requires: Pascal

Screen Unit **$45**
Terminal I/O unit a must for your Pascal library
- Device independent
- Field oriented with optional defaults
- Built-in editor (insert, replace, delete, clear, etc.)
- Requires: Pascal
- Source & object code included
- Input checking prevents user error
- No licensing fee for commercial use

All software unprotected

Circle 35 on Reader Service Card

What do over 20,000 Apple owners have in common? They are using **DATA CAPTURE**™ as their modem program. And time after time they recommended it to their friends. Or the company they work for suggests they buy it. And they voted it #31 of all Apple software in the 1982 Softalk reader poll. And it has been on the Softalk best seller list since 1981. If you are buying a modem program and you don't take a look at **DATA CAPTURE**™ you could be cheating yourself. Maybe there is user friendly modem software. And a 'friend of the user' company to stand behind it.

NEW **DATA CAPTURE**™ **5.0**
for Apple][and Apple][Plus owners

DATA CAPTURE™ **//e**
for Apple //e owners

NEW **DATA CAPTURE**™ **///**
for Apple /// owners

- **300/1200 Baud Operation** - for use with all popular information and timesharing systems plus other microcomputers, minicomputers, and mainframes.
- **40/80 Column** display on //e. 40 column or Videx display on Apple][with **DATA CAPTURE**™ **5.0**.
- **Edit Feature** - for preparing text to send to another computer or for editing text you have received from another computer. Many **DATA CAPTURE**™ owners use the editing feature off-line for preparing letters and/or performing other short typing tasks.
- **Capture Buffer** - You can List, Save, Print, or Transmit all or part of the Capture Buffer. The most flexible use of captured data of any program on the market.
- **One Key Commands** - output logon messages, passwords, and system commands with a single keystroke.
- **Menu Selections** - You don't have to learn complicated commands used by other types of modem programs. (Once you become accustomed to the menus you can skip them by using the Fast Menu feature.)
- **NOT Copy Protected** - We trust you.
- **Buyer Support** - We are a small company and we depend on your good will for our business. We have a five year history of full support for purchasers of our programs.
- Not sure what modem hardware to buy? - Call us for advice. For over 5 years we have been helping Apple owners put together the right communications software and hardware.

Ask your dealer for **DATA CAPTURE**™ or buy direct. We usually ship the same day and you do not have to wait for your personal check to clear.
We accept Visa and MasterCard.

Price - **$90.00.**

Software designed with you in mind.

Southeastern Software

7743 Briarwood Drive
New Orleans, LA 70128

504/246-8438
504/246-7937

DATA CAPTURE 5.0, **DATA CAPTURE //e**, and **DATA CAPTURE ///** are trademarks of Southeastern Software. Apple][, Apple][Plus, Apple //e, and Apple /// are trademarks of Apple Computer Corporation.

Circle 27 on Reader Service Card

on the values in the registers. When HPOSN returns, TEMPPTR, BYTENUM, and POSBIT are all set up. The current color mask is loaded from SHFTCHK and the Y register is loaded with the value of BYTENUM to be used as an offset. The color mask byte is exclusive-OR'd with the contents of the correct high-res screen memory byte, AND-ed with the pixel position mask (POSBIT), exclusive-OR'd with the high-res screen memory byte again, and the result is stored in screen memory.

In order to plot the point in the correct color, the color flag of the screen memory byte must be set to the same value as the color flag of the current color. This may affect the other six pixels in the byte, causing the well-known "bleeding" problem. To illustrate this, run the following short Applesoft program:

```
10 HGR: HCOLOR = 5
20 HPLOT 0,50 TO 279,50
30 HCOLOR = 1
40 HPLOT 100,50
```

The Demo Program

Listing 2 contains a demo program that lets you test out the new routines. You will have to BLOAD YTABLE, A$1B00 and then BLOAD the INIT and HGR routines we developed last time. Now you can BLOAD the routines in Listing 1 and you're all set to BRUN the program in Listing 2. It should draw two diagonal lines, one orange and one blue. They will appear dotted since certain colors cannot be seen in even columns and others cannot be seen in odd columns. Try modifying the program to draw white lines so that the lines look smoother.

Next Time

In the next column, we'll look at line-drawing algorithms and improve slightly on the one that Applesoft uses. After that, we'll be jumping into animation-quality block-shape routines. I will begin to give less detailed descriptions of how the machine language routines work since they do have some comments and we'll need to spend more time explaining the philosophy behind the implementation. Questions on past columns or ideas for future columns are always welcome. Adios till next time!

Hardware/Software Review

The Gibson Light Pen: An "Etch-a-Sketch" Graphics System for the Apple

Lee Webster

Gibson Light Pen *$350*
Gibson Laboratories, Inc.
23192 Verdugo Drive
Laguna Hills, CA 92653

JUST so you don't have to wonder or read the last paragraph first, let me state "up front" that this is a super graphics system! About two years ago, we purchased the Gibson Light Pen System at the office and it came with "preliminary" software. Nevertheless it was well done and we made a lot of use of the pen. In May, 1983 I attended the Applefest in Boston and saw the Gibson people demonstrating the "final" version of their software—it looked even better and I couldn't wait to get a copy. As a registered owner, we automatically got a copy (in late fall) for no charge. That's the kind of support you appreciate from a company!

Components of the System:

What do you get for your money? Hardware and software. They both come in a blue plastic box which looks like a disk storage box for 8-inch diskettes. Included are: the interface card with light pen attached, one disk of software (copyable so you can make your own back-ups), and a 5 x 8-inch, 104-page manual. The interface is mounted in what looks like a cassette tape box and is epoxied in so all you can see is the tab with the gold connections sticking out the bottom. (Maintenance? After the 6-month warranty period expires, you send it back to Gibson for repair or replacement for a flat fee of $55.) The software is packed on the disk (little room left) and is mostly in BASIC. It is the real heart of the system and I'll say more about it below. The manual is well-written, with graphics undoubtedly done with the system, and reproductions of screen displays. The Appendices have information on printer interfaces (PKASO, Super-MX, and Grappler are supported) along with command summaries for most of the application programs.

Installation:

To use the system, you need an Apple II, II+, //e, or ///, Franklin, Basis 108 or Bell & Howell computer with 64K RAM and a disk drive. For a monitor, any B&W or color will work, *except* Apple's Monitor ///, Amdek ///, and USI's PI 3. These won't work, according to the manual "...because they use long persistence phosphor." (I'll let you technical types figure out what that is all about!)

Installation is relatively simple, if slot 7 on your Apple is free. Otherwise there are some additional wires and connections to be made—carefully described in step-by-step fashion in the manual. Essentially, you just insert the interface card in slot 7 and route the cord to the light pen out the gap between the case and cover, just above the keyboard (not out the back). With the cover in place, this arrangement gives some stress protection to the cord, and also is more convenient as it gives you about 44-45 inches of free cord from the computer to the end of the pen. The pen looks like the casing from a felt-tip pen and contains a light sensor (no switches). The only difficulty I had installing it was due to the width of the interface card, since it is inside that plastic case. There is a tight fit in slot 7 next to the disk drive interface and it took a little re-arranging of disk cables to squeeze everything in.

Very simply, the pen is an input device. The light receptor at the end of the pen (not right at the tip but inset for protection) determines the location on the screen where you are pointing by sensing when the dot of light (used by the CRT to "paint" the screen image) passes it. To do this, the hardware keeps track of the "horizontal and vertical sync signal." Is that clear? You certainly don't need to understand it to use and enjoy the Gibson system!

Pentrak Software:

Now for the software—this is the guts of the Gibson Light Pen System. The hardware has worked flawlessly since we've had it (once we realized certain monitors would not work!). The new version of the software is *super!* The few drawbacks of the preliminary version (multi-step calibration, inconsistency among some applications programs in the way they did similar functions) have all but been eliminated. Overall, there are really two parts to the software. One is the Pentrak 2.0 machine-language system which gives access to the light pen system. The other is a series of excellent applications programs provided by Gibson Labs, written in BASIC, which are listable, modifiable, and which use the Pentrak features.

Let's take a quick look at Pentrak. Most users will want to skip this till later (as I have), but what it involves is providing about two-dozen statements accessible from BASIC by using the ampersand (&) hook. To show you what this looks like, here is a four-line program from p.9 of the manual:

```
10  & INIT, CHP, DHP, NEG
20  X = 140: Y = 92
30  & TRACK(X,Y,KEY,LINES)
40  & TP: PRINT "EVERYTHING
    OK"
```

It is used to check the installation of the system. The &INIT presets Pentrak options—CHP, DHP, and NEG to clear the Hi-Res page, specify which Hi-Res page is to be used, and negate the screen image. The &TRACK generates a tracking cursor for the pen. The arguments in parentheses deal with the horizontal and vertical position of the pen, the mode for terminating Pentrak function, and the generation of horizontal and vertical dotted flashing lines through the cursor (what I call "cross hairs") to help position the cursor. The &TP command selects the primary text screen mode.

Applications Software:

For most users, at least in the beginning, the application software provided by Gibson Labs is where you'll start. For many, they contain all you need, but if you want to make changes or write your own software, a good guide would be to list and study the programs provided. In fact, some routines (like the MENU program) are easily identified and usable in your own programs.

Let's get started with a trip through the system, using the applications programs. To begin, you need to boot the disk in drive 1, which automatically loads the Pentrak routines, and brings up the main menu displaying 12 items in two columns. The first thing you need to do, however, is to calibrate the system. By pressing C on the keyboard, you get brief instructions and a display containing a horizontal white bar and a vertical dotted line. (Pointing to a menu item at this point probably won't get you anywhere since the pen hasn't been calibrated.)

When you point the pen at the screen, the dotted line should follow (track) as you move the pen left and right. If it doesn't, you type a "(" or ")" to move the dotted line left or right until it falls under the pen and tracks properly. It is easy to do, and only takes a few seconds. It may be necessary to adjust the brightness or contrast on your monitor to get the tracking correct. When you press RETURN, you are back to the main menu.

The menu looks fairly ordinary, except that to the left of each item is a white box (about two or three times as large as the familiar Apple cursor). By pointing to the box next to the selection of your choice, the pen "reads" the position and issues a "boop" (as the manual calls it) to let you know that it has gathered the data. This audible feedback is a useful feature. Next, at the bottom of the screen, two larger white blocks are displayed. The one on the left has the word ABORT printed in it, and the other block contains the name of the program you selected. This gives you a chance to "bail out" if you selected the wrong program, but if everything is OK, you just point to the right-hand box and the program is loaded. This menu method is a real help for those who don't type well!

New DAVID-DOS II™ Compared With Other DOS's

Now you can speedup Text-file handling two ways

DAVID-DOS II is a new edition of DAVID-DOS with added speed, commands, and features. New Read, Write and Save routines are high speed. DAVID-DOS II updates full disks like Apple's Master Create. (The programs on your disks are not touched). DAVID-DOS II Inits blank disks with Basic, Binary or Exec HELLO in seconds. Ten new commands operate identical to existing DOS commands. Use them from the keyboard or in Basic programs. They accept A & L parameters.

Ten New DOS Commands

1. **TLOAD speed loads all Text Files, random or sequential, to ram.**
2. **TSAVE speed saves all Text Files, random or sequential, from ram.**
3. **TLIST Lists all Text Files, random or sequential to screen/printer.**
4. **DUMP Memory to screen/printer in Hex with Ascii on right side.**
5. **DISA disassembles Binary to screen or printer.**
6. **AL prints last loaded program Address & Length in decimal & hex.**
7. **HIDOS moves DOS to Language Card & continues operation of program.**
8. **/ is a one keystroke Catalog in addition to the original command.**
9. **DATE prints with any clock. Also File Dating with clock or manual.**
10. **FIND prints address's of hex found in 64k memory. Hidos cmd only.**

Compatible

All DOS entry addresses have been preserved. DOS is original length and compatible with most software. David-Dos II is copyable and creates fully copyable updated disks. DAVID-DOS II is licensed by programmers for inclusion in the software they sell. Init areas were used for David-Dos II. Works with all Apple IIs including IIe 80 Col, Franklin & Hard Disks, such as Corvus & Xebec. Requires 48K. Complete documentation for screen or printing and many utilities are on the disk.

All times in seconds. (Time Test programs available)		DAVID DOS-II	ProDOS	DIVERSI DOS	DOS 3.3
TEXTFILES (100 Sectors)	TSAVE	8.0	NO	NO	NO
	TLOAD	6.2	NO	NO	NO
(791 Strings,	WRITE	29.3	28.0	29.4	88.4
32 chars ea)	READ	24.3	16.3	24.3	83.8
	PRINT/READ	44.2	45.9	45.1	117.1
(442 Sectors, 7 x 500)	APPEND	142.3	142.9	151.1	1231.2
APPLESOFT (100 Sectors)	*SAVE	7.1	16.4	6.4	33.1
	LOAD	5.0	4.0	5.0	23.5
INTEGER (100 Sectors)	*SAVE	7.3	NO	6.6	33.4
	LOAD	4.9	NO	4.9	23.4
BINARY (100 Sectors)	*BSAVE	7.8	18.4	7.3	28.7
	BLOAD	5.8	4.8	5.8	24.5
48K PROGRAM SPACE	APPLESOFT	36,352	NO	36,352	36,352
(With 3 Bufs avail)	INTEGER	36,352	NO	36,352	36,352
	BINARY	36,352	34,816	36,352	36,352
64K PROGRAM SPACE	APPLESOFT	46,592	31,232	45,658	35,162
(With 5 Bufs avail)	INTEGER	46,592	NO	35,162	35,162
	BINARY	46,592	40,704	45,658	35,162
NUMBER OF DOS COMMANDS		37	29	31	28
CLOCK FILE DATING		YES	YES	NO	NO
MANUAL FILE DATING		YES	NO	NO	NO
ONE KEYSTROKE CATALOG		YES	NO	YES	NO
AUTO USE INTEGER CARD ANY SLOT		YES	NO	NO	NO

Each Program was tested twice W/Apple Clock Card on a newly formatted disk containing DOS.
*Add 5 seconds for Verify. Apple II, Applesoft & ProDOS are trademarks of Apple Computer

To Order: Send Check or Phone Visa/MasterCard.
Add $2.00 Shipping. Overseas add US $4.00
Calif. add 6%. All orders airmailed.

$39.95

DAVID DATA™

12021 WILSHIRE BLVD., SUITE 212A
LOS ANGELES, CA 90025 (213) 478-7865

Circle 30 on Reader Service Card

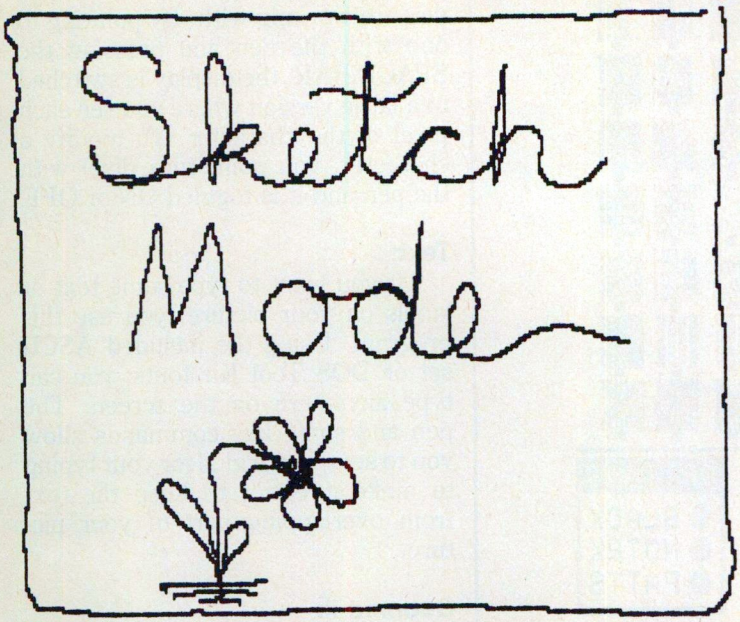


Figure 1

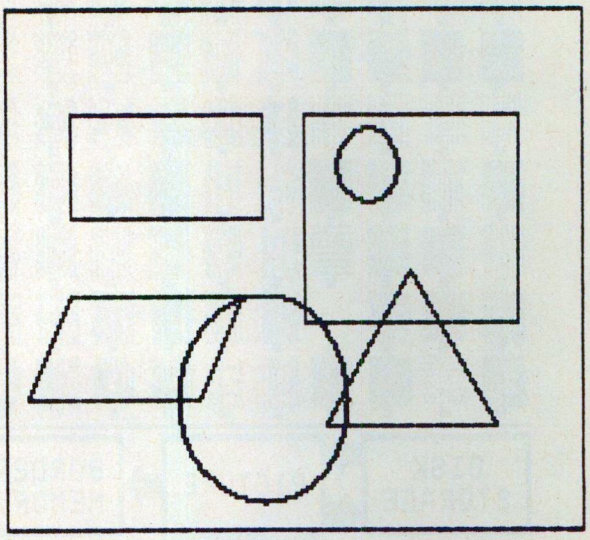


Figure 2

Now I'll take you through the main application programs provided. (On the main menu there is also a RETURN TO BASIC choice, and a PRINTER choice—PKASO and Grappler+ are included).

Sketch:

This program is used for free-hand drawings. As with most of the application programs supplied on the disk, it begins with a single text screen of HELP information reviewing the commands and procedures. You can return to the Help screen by pressing H. At the bottom of the screen there is a white bar containing the instruction to point to it when you are ready to continue.

To get started you can choose from among six colors by pressing a single key (e.g. D=black, B=blue, etc.). You then use a pair of keys to do your sketching. The "1" or "←" key starts you drawing a line, and the "2" or "→" key stops. (This is similar to the PENDOWN, PENUP sequence in Logo in that you have to actuate the pen before you can draw something.) Having a choice of key pairs for one function is handy so both right- and left-handers can be comfortable. (Remember, one hand also holds the pen!). After you've selected a color and pressed the key, moving the pen draws a line on the screen. To move to another location, you could pull the pen way from the screen (not the most precise method) or toggle the pen OFF with a "2" or "→" key. Now you can freely move to another location and start to draw another line. You can change colors at any time.

You also have the option of using the mirror feature, which allows you to draw on one half of the screen and have a mirror image automatically drawn on the other half. You can negate (reverse black to white, etc.) by using the negate command.

The ESC key takes you to a Filer routine—a routine that is common to several programs. A menu (like the main menu) is displayed and you select as before. Options available are: Load (a picture file), Save, Catalog, Negate, Help, Return to Menu, Erase (the picture), Return to Program, and Change System Configuration. The latter is used to set the slot and drive number for the data disk. (Since each program runs independently, you have to reset the configuration for each application program when you run it.)

To prove that I'm no artist, see Figure 1 which was created using the SKETCH program. As I don't have a PKASO or Grappler available, I simply took the binary files of the Hi-Res screen, and used a screen dump program to print them out on a dot-matrix printer.

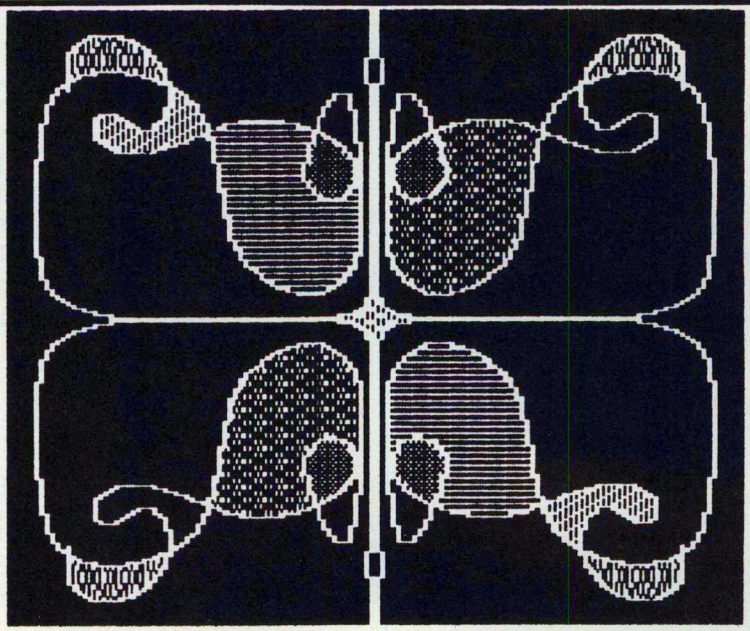


Figure 3

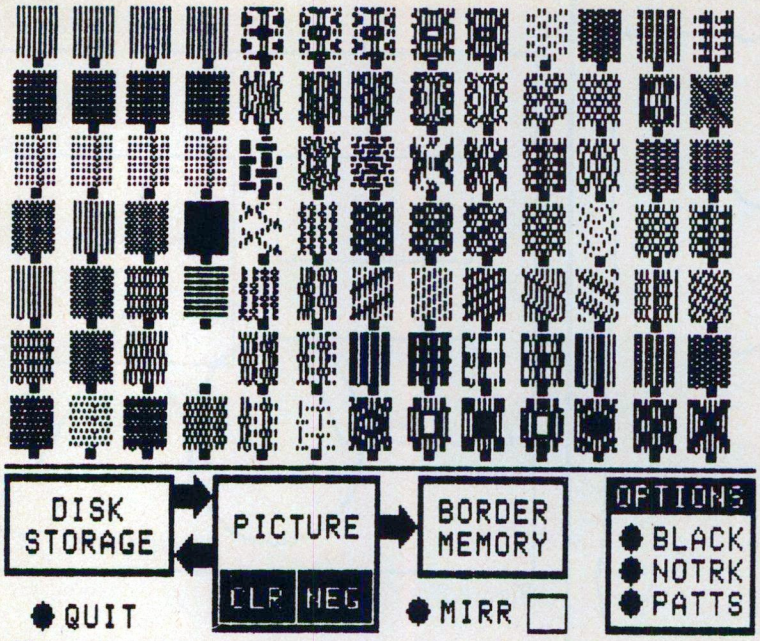


Figure 4

Boxes:

When you return to the main menu, in most cases the picture remains in memory and you can work on it with another program. I did notice that I usually had to recalibrate the pen when returning to the main menu—this was true when using either a B&W or color monitor.

The BOXES program draws rectangles and then shades two sides to create a 3-D effect. To use it you first point the pen at the location where you want one corner of the rectangle and press a key. As an aid in locating the rectangle, the cursor has a set of "cross-hairs" through it—a vertical and horizontal dotted line. Moving the pen to another location on the screen causes a "rubber band" or temporary rectangle to be drawn—it stretches as you move the pen so you can see what your rectangle will look like. When you get the other corner where you want it, press another key and it is drawn and the shading added automatically.

You can also add text in the box after it is drawn. You just start typing and the text is automatically centered in the box. If the box is large enough, you can have more than one line of text.

Font Editor:

When using this program, you lose from memory the last picture you were drawing—but with the Abort feature on the main menu you can go back to use the Filer to save the previous picture first if you need to. The purpose of FONT EDITOR is to create or edit the standard ASCII character set or the DOS Tool Kit sets. It is limited to working with 7 wide x 8 high fonts.

The menu of options uses a different format. A series of five boxes (each containing a one-word label) is overlaid on the bottom of the screen. To select one, you point to the boxes with the pen. The Save, Load, Menu, and Help options are obvious.

When you select Edit, the real fun begins. The screen display changes to show the complete character set you are working with. By pointing to one with the pen and pressing the SPACE BAR, the display is switched to a large version where you see each pixel of the character. To modify a character, you point to a pixel with the pen and it is toggled ON or OFF.

Text:

If you want to type some text or labels on your picture, you use this program. Using the included ASCII set or DOS Tool Kit fonts, you can type anywhere on the screen. The pen and some key commands allow you to set the margins for your typing to make it easier to keep the text from overlapping part of your picture.

Geomed II:

This program is perfect for drawing standard geometric shapes. You select the shape type by pointing to an overlaid menu of icons (dare I call them that?!) along the top of the screen. There are icons for straight lines, rectangles, parallelograms, triangles, circles, and a choice of black or white lines. To access the Filer routines, you point to the icon that looks like a diskette! After you point to the icon (and it registers with a "boop"), the menu disappears from view. You return to the menu by pressing ESC.

If you want to draw a rectangle, the method is similar to using BOXES, except there is no 3-D shad-

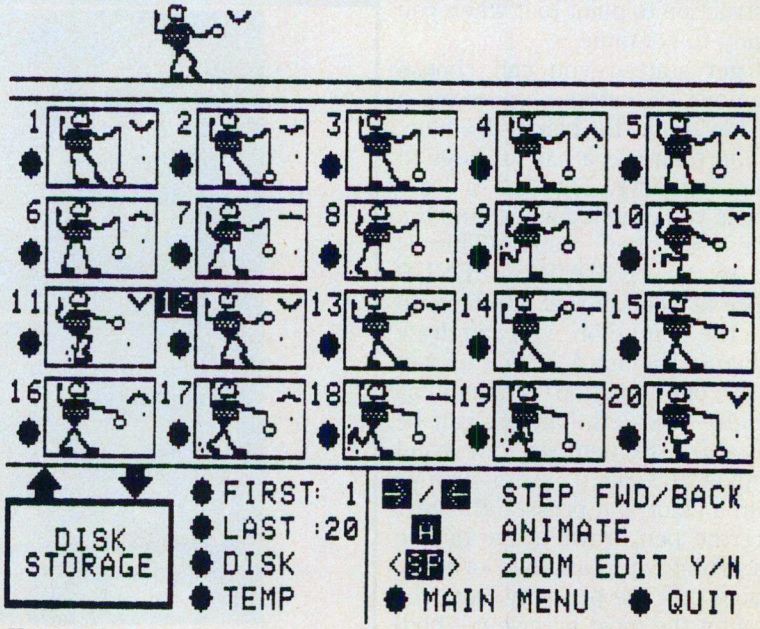


Figure 5

ing. Some shapes require you to enter three defining points—for example to get a triangle you define the line for one side and then the point for the opposite vertex. All shapes have the "rubber band" feature so you can easily see what it will look like before finalizing the shape. When drawing a circle, you point to the location for the center, press a key, then as you move the pen, a hexagonal approximation of the circle is drawn. When your pen reaches the point at the end of the radius defining your circle, a press of another key causes the program to replace the hexagon with a more precisely drawn circle. The only problem I had with this program was that the light pen sometimes balks when used near the screen border (doesn't register as quickly) and that is where the menu is overlaid. Pointing again or adjusting the brightness/contrast on the monitor usually solves the problem.

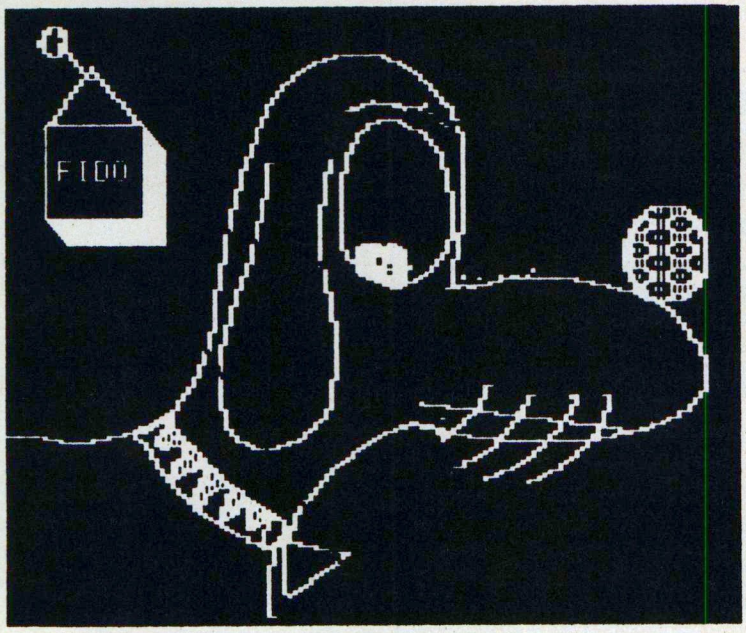


Figure 6

This program provides an easy way to create flow diagrams and any other drawing requiring geometric shapes. My crude example is shown in Figure 2.

Grid Draw:

There are times when you want your lines to be automatically vertical or horizontal and you want the corners and diagonals to match properly. GRID DRAW helps with that. You can only draw straight lines, in six colors, but all lines will start or end on a grid point. These grid points are a series of dots plotted on the screen.

GRID DRAW also contains a sample of Pentrak's "screen lifter" routines. By pointing the pen to the far right of the screen, the tracking ceases. If you move the pen up, the drawing screen is raised like a curtain, with the Help screen appearing underneath!

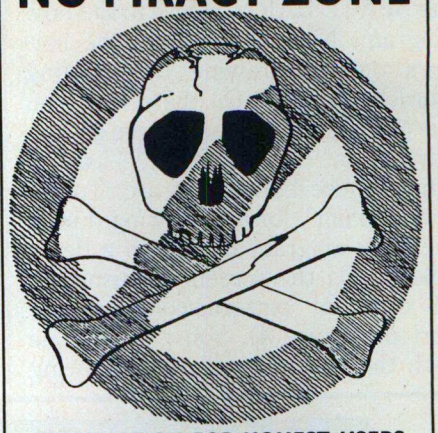
Penpainter:

This is one of my favorites—it's great for you budding Michelangelos! As you can see by my example in Figure 3, I'll never make it! Essentially the program lets you "paint" or fill in any outline shapes on your drawing using a selection of colors and patterns from a palette.

When you run the program, the bottom of the screen displays the third type of menu seen in these application packages. (Remember the other two?—text menu, and overlaid icons.) In this case, there is a flow diagram shown at the bottom of Figure 4. To select a function, you point to the block or the arrow.

The way I did the drawing in Figure 3 was to first set the mirror option so that when I drew in one quadrant, a mirror image was automatically shown in the other 3 quadrants. I then pointed to the Picture block on the diagram and was presented with the drawing screen. I just drew some lines (in a fashion

NO PIRACY ZONE



INFORMATION FOR HONEST USERS

Thousands of Apple users have already joined us.

Hardcore COMPUTIST

What you can't get anywhere else.

- Techniques to unlock software
- Unlocking tutorials for beginners
- Tips on program modification
- How to modify DOS
- Game secrets PLUS Advanced Playing Techniques (APT)-how to get those extra ships and weapons
- In-depth product reviews
- Straight answers to your questions

If you're a vigorous Apple computist,

YOU CAN'T AFFORD TO BE WITHOUT US ANY LONGER!

Annual Subscription Rates
Please check one of the following:

☐ U.S. $25
☐ Canada, APO/FPO, 1st Class $34
☐ Mexico $39
☐ Foreign airmail $60
☐ Foreign surface mail $40

☐ Yes, start my subscription now.

H1

Name _____
Address _____
City _____ State _____ Zip _____
Country _____
Visa/MC _____ Exp _____
Signature _____

Send to: Hardcore COMPUTIST
Subscription Department
P.O. Box 44549H
Tacoma, WA 98444

Circle 9 on Reader Service Card

similar to SKETCH), then pressed the SPACE BAR to return to the flow diagram. Next, I "locked" the borders on the drawing by selecting Border Memory. On the screen you see the lines and screen edges getting darker as the computer "learns" where these boundaries are. To get some "paint" for our drawing, I used the flow diagram to call up a demo palette. On the screen are displayed 91 different patterns/colors (upper part of Figure 4). You point to one with the pen and when the drawing returns to the screen, you point to the enclosed area you want "colored in" and it is automatically filled in! It is fascinating to watch as it colors in all the nooks and crannies.

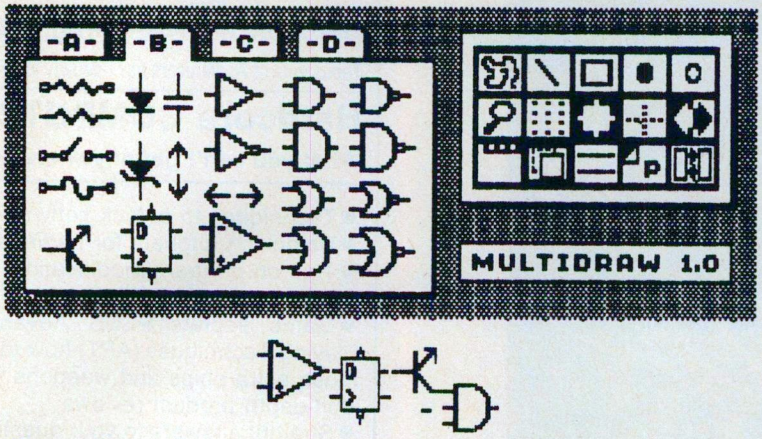


Figure 7

If you make a mistake, you can color in white (same color as the background). Occasionally your "boundaries" for an area are not continuous and the color "leaks out" of a gap in the border and paints the whole screen! You can, of course, paint pictures you've created using the other programs, as long as you do the Border Memory step first. The end results with this program are spectacular on a good color monitor!

Pattern Editor:

So you don't like those 91 patterns on the demo disk? This program allows you to create some of your own or modify existing ones. A flow diagram menu at the bottom of the screen gets you between Disk Storage, Palette Screen (where the 91 are shown), and the Pattern Editor.

When you move to the Pattern Editor, the upper left part of the screen shows the pattern design repeated in a 5 x 6 grid so you can see what it will be like when filling an area. The right 2/3 of the screen shows a large scale view of the pattern where each pixel is accessible. Menu choices in the lower left part of the screen allow you to choose to edit by point, line or the whole screen. You also choose one of six colors. Pointing to a particular pixel on the large-scale display "paints" the desired color. You can really be creative although a color monitor is a must for this work.

Animator:

If having animated sequences in your programs is "your thing," this program is for you. It permits you to create/edit up to 20 frames of your animated sequence. Figure 5 shows the display for the Gibson Labs demo. The display area at the top of the screen shows you the animation in real-time if you choose automatic mode. You can also step through the sequence one frame at a time backwards or forwards, while the particular frame being used is highlighted.

The toggle features (your choice is confirmed with a click) include: choice of black or white lines, whether to have the cross-hairs visible, whether to have the cursor visible, visible grid dots or not, and which portion of the drawing screen you want shown at the bottom half of the screen (this allows you to have the icon menu and Card Deck on the screen simultaneously with the upper or lower half of the drawing). You can press the SPACE BAR to remove the upper half display completely—another press overlays it again.

When you point to any of the other icons, the border around it flashes to confirm your selection. These options include lines, rectangles and sketching, which each work as previously described, including the rubber band feature. The printer option takes you to the routines for dumping your graphics. The Filer icon (not too clear—a diagram with two blocks and arrows between them) overlays a filing flow diagram on your picture. (See Figure 8) This is a fancy version of the flow diagram menu used on some other programs. When you point with the pen to one of the blocks or the arrows, the display reverses (like CARDS and GO!). "Drive" toggles between 1 and 2, and "Slot" between 4,5,6 and 7 each time you point to it. Finished selecting? Point to the GO! block to execute your choice. The X block removes the Filer menu and returns you to the drawing screen.

The most useful feature, however, is being able to create or edit a frame. By choosing the Zoom mode, you can select which frame you want to fill the screen in a 40 x 24 display. You can toggle any pixel ON/OFF by pointing to it with the pen. Just for something to do, I modified frames 1 and 20 so the yo-yo got squished on the bottom of its travel—all very easy to do, even if not significant!

As mentioned earlier, all these application programs can work in concert on a single drawing. As an example, see Figure 6 (it is supposed to be a dog!!). I used the SKETCH program to do the outline of the pooch, the BOXES routine to put the shadow picture with title on the wall, and the GEOMED II for the circular nose, the mouth, eyebrows, triangular tag, nail and nail head. Finally, PENPAINTER was used to put the pattern on the nose and in the collar. Certainly no award winning drawing!!

Multi-Draw

OK, I admit it—I saved the best for last! It is Gibson Light Pen System's answer to Lisa Draw! There are a couple of features that really make this a stand-out program.

Let's begin by looking at the screen layout (Figure 7). On the right side of the upper half of the screen (looks a lot like a Lisa display!) is a table of 15 icons. The left side is for the Card Deck (more about that later). Some of the icons merely tog-

gle a particular feature, others get you into a particular option.

One super feature is the Zoom Mode. After selecting this from the icon menu, you point to a location on the screen drawing and press Z or ?. Immediately the screen is filled with a 40 x 24 display of that area of the screen. You can then toggle each pixel ON/OFF, and even scroll around the screen while in Zoom! This is invaluable for cleaning up those little messes that occur when drawing (not getting things lined up properly, moving the pen too fast so it skips in drawing a line, etc.).

The most powerful feature is the Frame Mode. This allows you to copy parts of a picture to other parts of the screen. Here's how it works: you select the area of the screen you want to copy by pointing to one corner of the area (cross-hairs mark the spot), then point to the diagonally opposite corner of the area you want to copy and press the key command twice. Now when you move the pen, the whole area previously identified moves with the pen! Pressing another key locks the copied portion on the screen. An additional feature is that you can stretch or shorten an image, and you work in the Zoom mode too.

Where this really comes in handy, is with the Card Deck. These are a set of frequently used symbols, letters, logos, etc. that you can save and load when needed. When loaded, they are displayed in the upper left corner of Figure 7. There are four cards per deck and you select which card you want by pointing to the tab—card B in this example. You can use the Frame Mode to "pick up" a symbol off the card and "float" it onto your drawing. You can also do the reverse to add to your card deck. I used these cards to do Figure 9—you can see I am no electrical engineer! (How did that bath tub get in the circuit?)

Conclusion:

The Gibson Light Pen System is an excellent implementation, well designed, with good documentation (you really don't need to read it much except for installation and to program using Pentrak), which has worked flawlessly so far. As an indication of its "user friendliness," my 12-year-old son was watching me work with the pen. Later I let him try it alone, and he made out fine just working with the Help screen instructions (not reading the manual). Whether a light pen, a mouse, touch screen, or graphics tablet is the best way for graphics input depends on your application and your own opinion. I did notice that the old arm does get weary reaching and holding the pen up to the screen during long sessions—maybe I just need to get into shape!

Check this system out! I think you'll be impressed with what it does and how easy it is to use. The kids toy "Etch-A-Sketch" was never like this!!

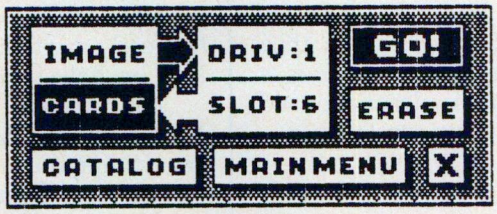


Figure 8

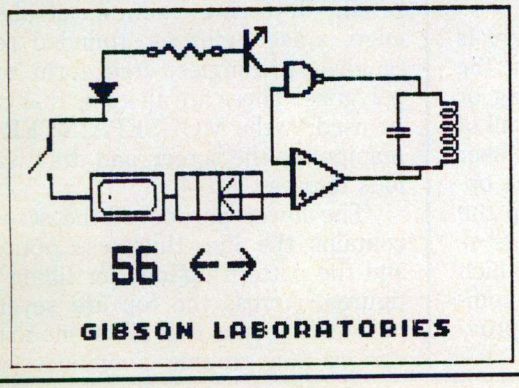


Figure 9

FREE?DISKETTES

SAVE MONEY! Apple II+/e users can use the diskette flip side, if another "write enable" notch is correctly made.

The DISK-NOTCHER by QUORUM quickly SOLVES that PROBLEM. It's like FREE DISKETTES!

- Stainless Steel Guide
- Easy Leverage Handle
- Clippings Catcher
- Square Notch Cut
- Black Finish
- Get THE BEST!

Certifix™

BE SAFE! Your 'FREE' disk is CERTIFIED 100% ERROR FREE with CERTIFIX by QUORUM. It 'LOCKS OUT' DISK FLAWS and lets you use the rest. Displays status report & saves it to disk. Next, CERTIFIX automatically formats then offers to initialize with genuine Apple DOS 3.3 too. Great for testing economy disks. CERTIFY, FIX & INITIALIZE every disk with CERTIFIX!

100% Money Back Satisfaction Guarantee!

DISK NOTCHER is $14.95
CERTIFIX™ is just $24.95
ONLY $29.95 for BOTH!
Add $1.50 s/h • CA add 6½ % tax

QUORUM INTERNATIONAL, Unltd.
INDUSTRIAL STATION P.O. BOX 2134-CA
OAKLAND, CA 94614

Circle 11 on Reader Service Card

"TELEDELIVERY"?

- SOFTWARE ON DEMAND
- FASTEST DELIVERY ANYWHERE
- 24-HOUR ACCESS, WORLDWIDE
- SAVE TIME, TRAVEL, MONEY
- EXCELLENT CUSTOMER SUPPORT

TRY IT NOW!

CALL, VIA MODEM: (213) 516-9432
AND EXPERIENCE THE FUTURE.

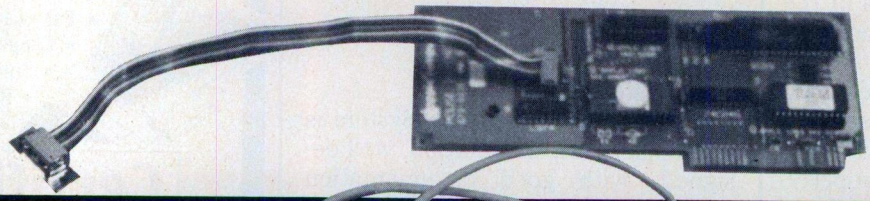
TELEPHONE SOFTWARE CONNECTION® INC

POST OFFICE BOX 6548 - TORRANCE, CALIFORNIA 90504

Circle 6 on Reader Service Card

Hardware Review

The Apple Mouse //



John Cox

THERE are a lot of good apples in the barrel these days, and at first glance, it certainly would seem that the good old Apple][is rapidly going to the bottom. Lisa 2 and Macintosh are wonderful machines with more features than we might think possible. The mouse interface is very nice to use and really should be experienced to get the feel of it. But what about a person, like the author, who has invested a great deal of time and money in the Apple][family of computers? Well, there is really good news coming from Apple these days. We are constantly receiving information and products from Apple pointing out to us that the Apple][is not dead in the water. The Apple][is moving right in step with the other developments that Apple Computer, Inc. is implementing on the newer machines.

The most playful addition to the Apple][is the AppleMouse][. This adds the pointer control that is intrinsic in the design of the Lisa and Macintosh computers. With the addition of these capabilities to the Apple][, we will get a great deal of power and flexibility in the user interface area. The mouse comes with enough information to use it in 6502 assembly language and in Applesoft Basic.

The documentation is a 56 page booklet containing very clear instructions on installing the mouse circuit board and the connectors for both the Apple][+ and the *//e* (see Figures 1 and 2). The writing style is geared to the level of a computer user rather than a programmer, so we will call it a fairly *idiot proof* booklet. (Computer users please take no offense by the use of this term.) The book describes the companion program MOUSEPAINT as well as the nitty gritty details of using the mouse in BASIC and 6502 programs.

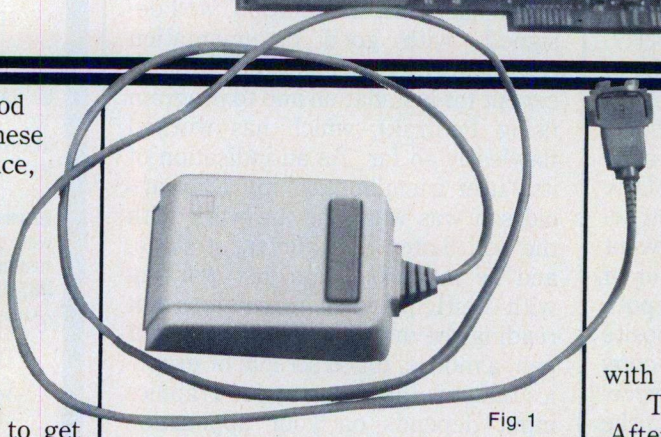


Fig. 1

The mouse is simple to use from a BASIC program, and will be of great benefit in menu selection and choice of options in programs that now use complex CONTROL or ESCAPE sequences. The entry points to the 6502 subroutines are provided so that the mouse may be initialized and its characteristics modified. Sample programs in both BASIC and 6502 are given at the end of the book.

Some of the descriptions of the MOUSEPAINT program in the manual are not made as clear as they might be. For example, there is a GRID option that is described and contains the following sentence:

> It keeps working until you tell it to stop.

Not a bad sentence, except there is no indication in the book or in the program that the grid is working or how to make it change. Almost all of the other options at least tell the user to click the mouse button on the option to toggle it. This brings up the complaint that there are no flags or checkmarks telling the user which option is in effect on most of the command options. This could be improved.

The AppleMouse][comes with a fabulous offering called MOUSEPAINT which was developed by Bill Budge and is based on Bill Atkinson's MACPAINT program. Most of the features of the MACPAINT software are transported to the lower resolution Apple][with great care, perhaps genius.

The program works like this: After plugging in the mouse circuit board and attaching the mouse to the connector in the back (see the photograph), you put in the MOUSEPAINT disk and boot it up. The options available at first are an interactive training program that teaches you to move the mouse and learn the jargon of "clicking" and "dragging," using "pull down menus" and various other mousey things. This whole concept carries directly through to the Lisa and the Macintosh interfaces. The other option is to go straight to the MOUSEPAINT program. This program allows you to draw on the high-res screen using the mouse, but to merely say that, is like saying that Niagara Falls is wet.

The options available to the MOUSEPAINTER are diagrammed in a column down the left side of the screen (see Figure 3). These are: hand, editors box, spray can, paint brush, lettering, pencil, straight edge, eraser, ellipses, rounded rectangles, rectangles, free form and polygons. These are all items that can be used by the MOUSEPAINTER to manipulate the screen and draw various figures.

The lower portion of the screen contains the line thickness options and the pattern options for filling in figures. Across the top are several pull down menus that allow one to do various non-graphic commands (see

Figures 4, 5, 6 and 7). Under FILE, are the options for saving and retrieving pictures and dealing with disks such as formatting a new disk. The EDIT option has commands to invert, delete, and flip horizontally and vertically anything that has been outlined by the editors option. The FONT menu contains the various fonts available for labeling the diagrams. Under AIDS, you may view the entire page (normally only the top left ¾ page is showing), set up an invisible grid to help make straight lines, work on a bit by bit level called FAT BITS, and select the type of paint brush for use under the paint brush option.

Now let's look at how this program is used. The mouse is used to move a pointer on the screen to a position on top of a word or picture (called an *icon* in mouse jargon), and the button is pressed to invoke that option. This is a noun–verb scheme of commands, while the classic command format has usually been a verb–noun scheme. So, using the mouse, we essentially say FILE OPEN instead of the usual OPEN FILE. This method allows you to think more naturally about computing because he can find what object you want to work with and then decide what it is you want to do.

To get a blank picture on the screen, you move the pointer to the FILE menu bar option and press the button. Then you drag the mouse toward you, which moves it down on the screen. This move shows the options allowed under FILE, one of which is to get a blank picture (or screen). After this, you are free to express yourself however you see fit, using the tools at your disposal.

The picture of the sunset at the seashore is a quick example of what you can do. If you want to draw freeform, you can use the pencil. For those who have tried their hand at art with a joystick, you will find this easier and more accurate with the mouse. The seashore and the reflections in the ocean were done using the pencil.

For shading things, you can use the spray can. The spray can is a fabulous idea and is well implemented. Just as in a real spray paint can, the more you spray back and forth, the more paint is left on the item. For a spray can example, see the clouds in the picture and the bushes on the left side.

The sun was created using the fill in the ellipse option, which works by filling a rectangular box with the largest elliptical shape possible without leaving the confines of the box. This is very easy to do because the circles are drawn interactively and you can "rubber band" them around until you are satisfied. The rocks on the beach were made using this option as well, with several different fill patterns chosen from the options at the base of the screen.

To create the ship, the polygon filling option was used with different patterns. The polygon option works

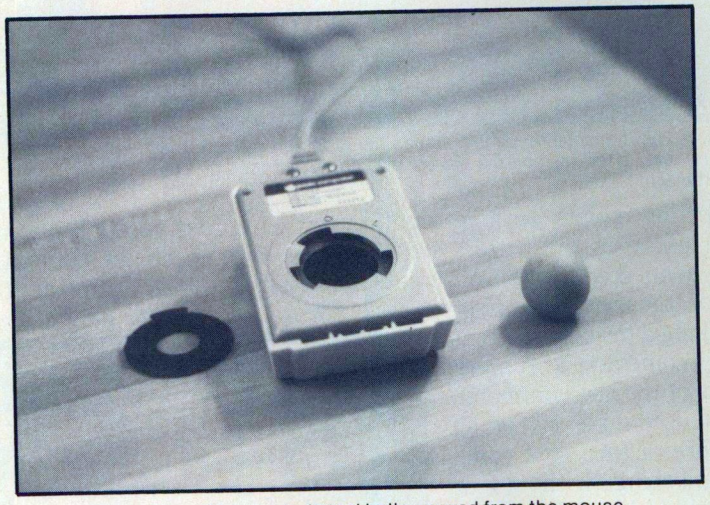


Fig. 2 The rubber coated steel ball removed from the mouse.

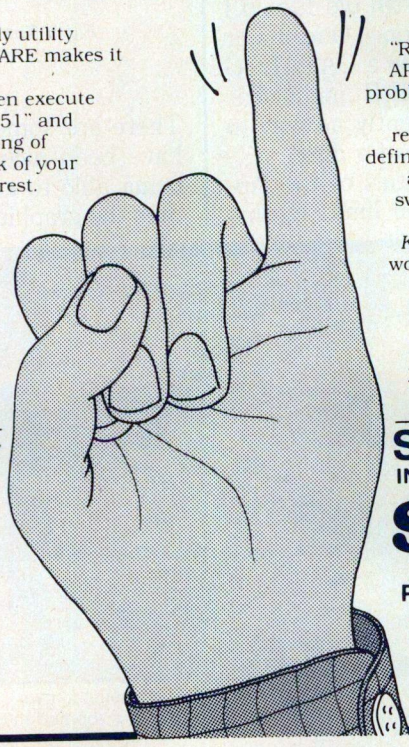
"CATALOG"

with the flick of your little finger!

KEYFLEX™ another handy utility program from TROY SOFTWARE makes it possible.

Now you can type and even execute "CATALOG", "LIST", "CALL-151" and any other instruction or string of characters with just the flick of your finger. *KEYFLEX*™ does the rest.

Any key on the keyboard can be defined by you to print and even execute any command or instruction. Why not define the "C" key as "CATALOG" and maybe add "L" to "LIST" your program.



Tired of typing all those "REM"s when writing your APPLESOFT programs? No problem, define it as "R". Any BASIC command can be reduced to one key. All key definitions are easily stored in a table on disk, and even swapped with other tables on disk.

KEYFLEX™ is designed to work with APPLESOFT and ProDOS.*

Order today and let *KEYFLEX*™ handle your repetitive inputs and let your little finger do more than it ever did.

Contact your dealer or order direct.
(Visa and Master Charge orders accepted)

Please add $2.50 for postage and handling.

TROY SOFTWARE DESIGN INC.
63 Roseview Avenue
Richmond Hill, Ontario
Canada L4C 1C6

☎(416) 884-4527

SPECIAL
INTRODUCTORY OFFER

$39.95

Please specify APPLE II or APPLE //e.

*ProDOS requires 64K RAM and APPLESOFT in ROM. APPLE and ProDOS are registered trademarks of Apple Computers Inc.

Circle 34 on Reader Service Card

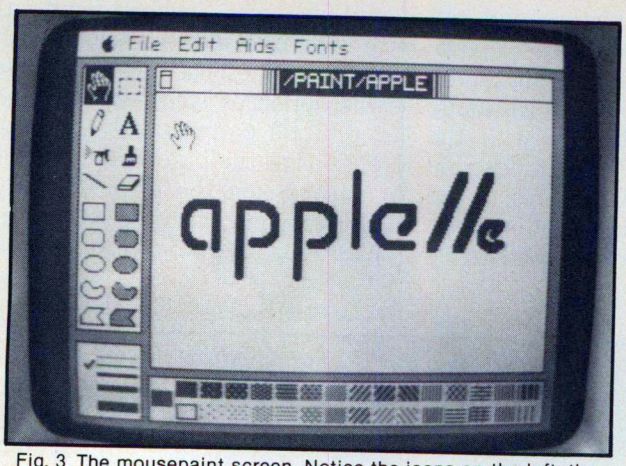


Fig. 3 The mousepaint screen. Notice the icons on the left, the menu across the top and the pattern and line selections across the bottom.

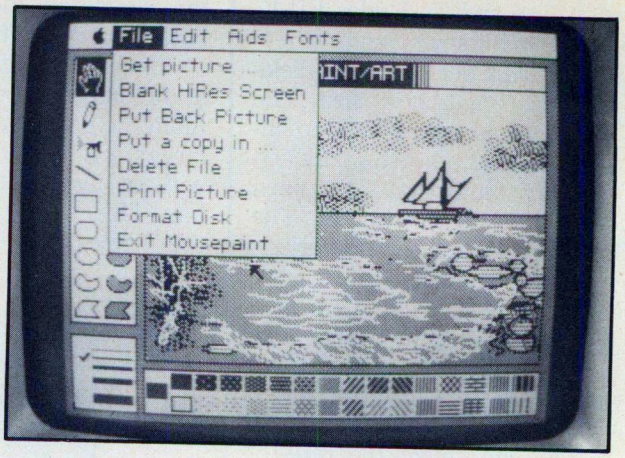


Fig. 4 The file menu options.

by connecting points with straight lines and filling in the pattern once you have connected to the first point again. For those who would worry about making a mistake, there is an UNDO command to remove the effects of the last command. This was used extensively on the seashore picture so I could try different things without fear of ruining what I had done up to that time.

The MOUSEPAINT program is well implemented and a joy to use. I did have one problem when I tried to save a file and the disk was full. The program somehow lost track of what was what, and therefore it wasn't able to delete an old file so I could save my new one. One major improvement would be a CATALOG option under the FILE menu bar listing. There is currently no way to examine the contents of a disk!

The graphics abilities of the Apple][are taken to their limits by this software, and you will be impressed with the speed of motion of the mouse and the graphics together.

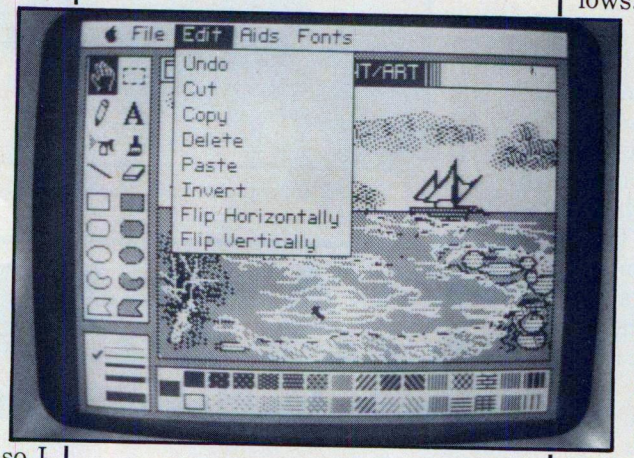


Fig. 5 The edit menu options.

There are some features that would have been nice if included, such as being able to create a picture larger than the graphics screen so you could fill a sheet of 8 ½ by 11 inch paper, and the ability to fill in an enclosed area as the MACPAINT program allows.

The AppleMouse][will retail for about $150.00 and will include the mouse and its controller board as well as the MOUSEPAINT software. The mouse needs to operate under ProDOS to take advantage of the interrupt capabilities of the new operating system. There is a set of procedures available to allow its use under Apple Pascal also. Once again, the mouse can be easily used in a BASIC program, but the performance is faster and better under machine language.

There are many other wonderful developments coming for the Apple *IIe* and mouse combination. The first is a new character generator ROM which includes many graphic characters that will facilitate the creation of Macintosh-like programs. The char-

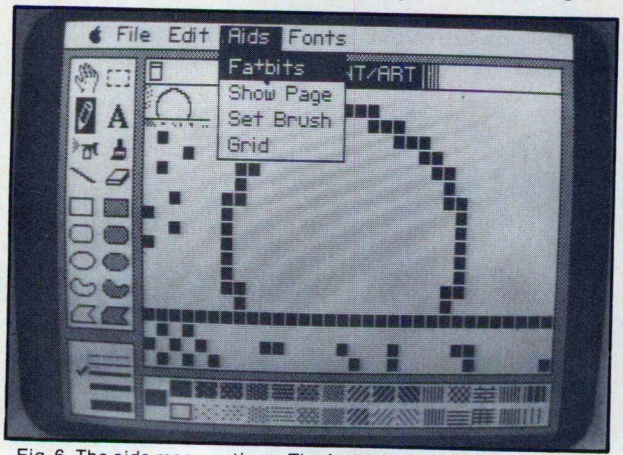


Fig. 6 The aids menu options. The fat bits option is being shown. The small area in the top left of the window is being zoomed in on for bit by bit access.

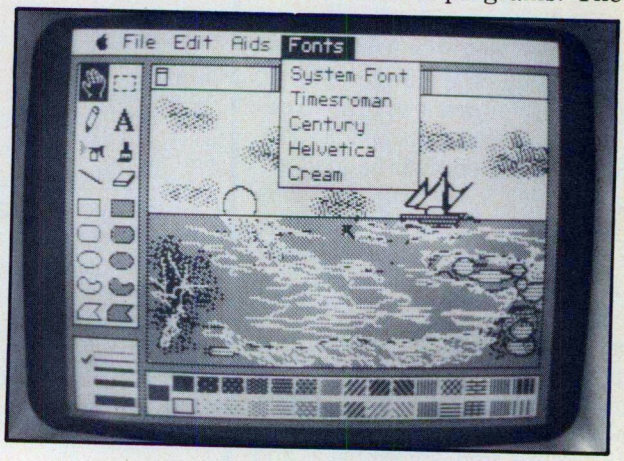
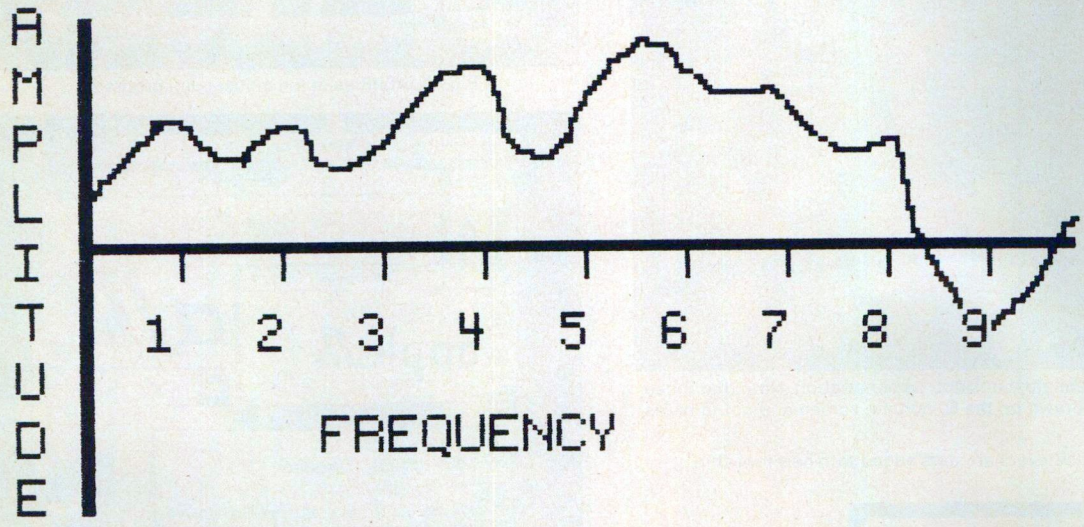
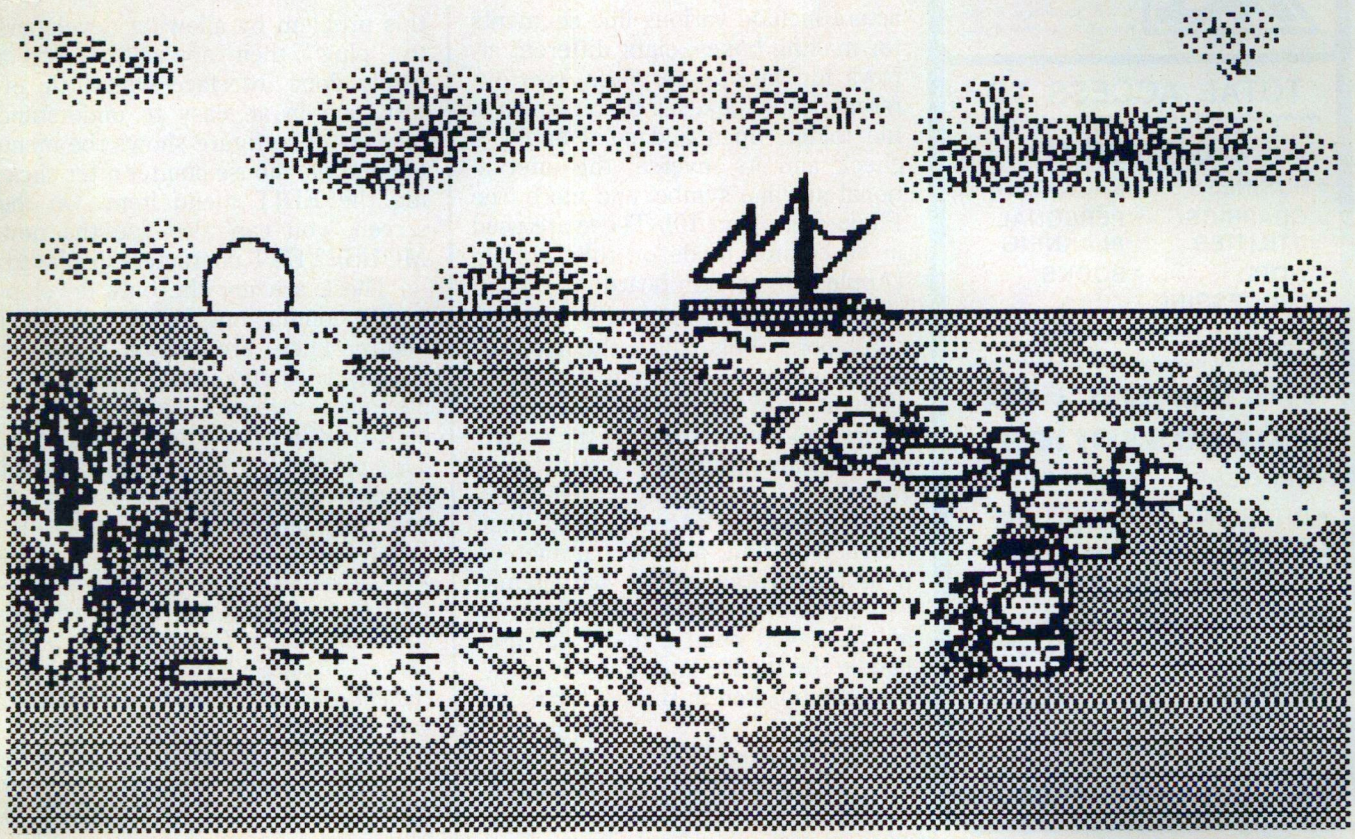


Fig. 7 The fonts menu options.

There are several different fonts.
There are several different fonts.
There are several different fonts.
There are several different fonts.
There are several different fonts.





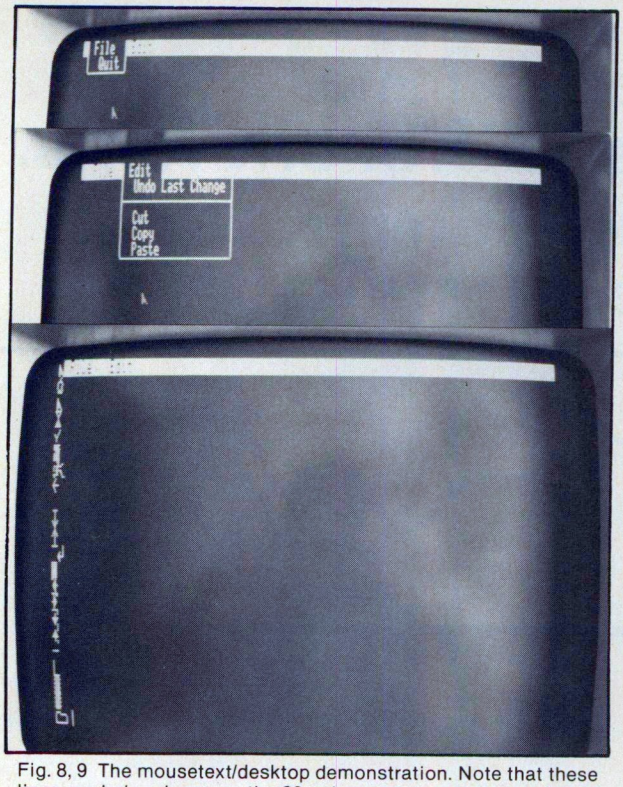


Fig. 8, 9 The mousetext/desktop demonstration. Note that these lines are being drawn on the 80 column screen and not in hi-res graphics!

Fig. 10 The mousetext characters added with new rom chip.

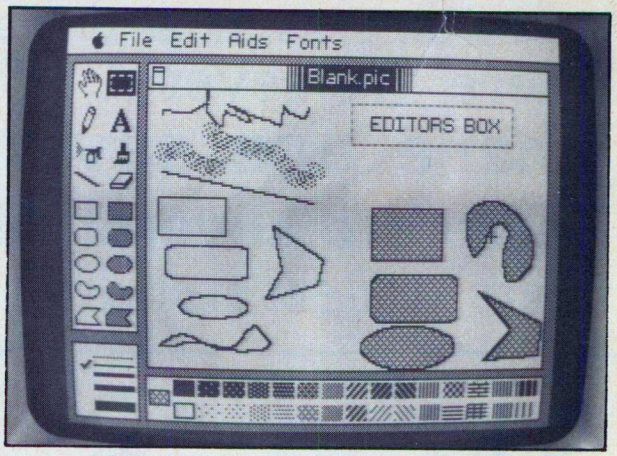


Fig. 11 Abilities of the mousepaint program.

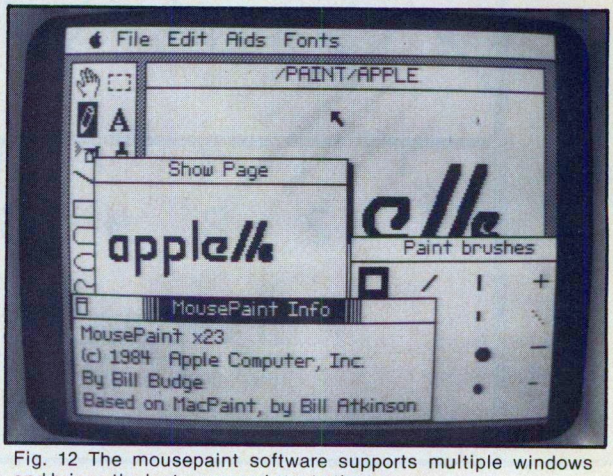


Fig. 12 The mousepaint software supports multiple windows and brings the last accessed one to the top.

ACCESS ALL MICRO
11306 Southland Road
Forest Park, Ohio 45240

TOTAL ACCESS TO
Software
- BUSINESS
- EDUCATIONAL
- GRAPHICS/ UTILITIES
- PERSONAL PLANNING
- WORD PROCESSING
- BOOKS
- ENTERTAINMENT

Hardware and Accessories
- PRINTERS
- DISK DRIVES
- INTERFACES
- MONITORS
- MODEMS
- DISKS
- RIBBONS

CALL
1-800-543-1114
Ohio · Alaska · Hawaii
Call 513-825-5803
CINCINNATI CUSTOMERS USE
OUR SOFTWARE PICK-UP FACILITY

Circle 15 on Reader Service Card

acters include various line segments for making boxes, eight different arrows for various purposes, two different dotted boxes, a running man, a file folder, open and solid apple, a check and its inverse, the international newline symbol and more (see Figures 8, 9 and 10). These are used in 80 column mode on the *IIe* only (Apple's 80 column board).

The next development is a set of developers' tools called the desktop toolkit which allows one to easily implement all of the new graphics characters and menu bars as if they were new machine language instructions. For example, to display a menu, you call the desktop with a code for that instruction and a few parameters pointing to the menu bar items, names, and attributes. The desktop software will build a menu bar and handle the mouse interface when the button is clicked in the proper menu item.

This is generally complicated software to implement and, unfortunately, the type of nice touch most software never gets. Apple has solved this problem by allowing developers to "plug" their application into a nicely done interface with little effort, and in an easy to understand manner. The figure shows the menu bar and the mouse pointer after clicking the EDIT menu item. On the screen, you can also see the new MOUSETEXT characters displayed.

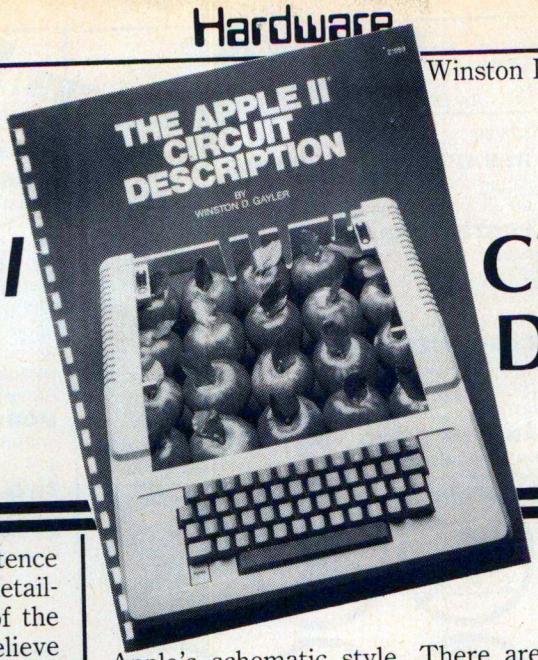
These are not the only developments coming out for the Apple][family. There are several that have already been announced such as Apple Pascal version 1.2 and the Profile hard disk interface for the Apple][. This old apple tree we've all grown accustomed to just keeps putting out new buds and blossoms all the time. What's next!

John T. Cox is a geophysicist currently contracting with Texaco Inc. He is the author of ASSYST: The Assembler System and the Thunder C Compiler, both available for the Apple][family. He may be contacted at the following address: Thunder Software, POB 31501, Houston, TX 77231 (713) 728-5501.

Book Review

Winston D. Gayler, Howard W. Sams & Co., Inc.

The Apple II Circuit Description



Leonard H. Anderson

THE first preface sentence reads, "This book is a detailed circuit description of the Apple][computer." Believe it! Available to members as order code "PACD" for $18.50 ($22.95 retail), this is *the* hardware reference for *all* Apple][main boxes up to the last revision before the *IIe*. If you don't know hardware, give one to a friend who does. Make certain your local repair center has this book or one with as much detail. It is that good.

This large-format (8½" x 11") book by Winston D. Gayler details all aspects of the motherboard and keyboard from Revisions 0 through 7 to D. Eight chapters, four appendices, plus 44 (!) "B-size" fold-outs explain all in detail. Schematics have been redrawn in the easy-to-read format of the old "Red Book." It is aimed at a reader ranging from serious hobbyist level to a working circuit-design engineer. There is a wealth of material in 172 pages of text. Text is to the knowledge range of the reader although a few sections tend to plod with "we see now . . ." phrases.

Gayler knows circuit design. This is apparent from the emphasis on timing diagrams and organization by function. Chapters begin with basic architecture and get into detail with the clock generator and sweep timing, video timing, memory (good detail on dynamic RAMs), on-board I/O and differences between the display modes. Illustrations are good, clearly drawn, and numerous. Several diagrams could have been moved to better fit descriptions, a minor fault common to many publications.

Text is comprehensive and instructional. Author Gayler uses Apple's signal line mnemonics (with explanations, a good point) and holds to Apple's schematic style. There are many references to one of the fold-out diagrams. This is a definite aid, but 44 fold-outs take up space. A suggestion for your own reading is copying most of the fold-outs despite the copyright warning; it saves later patching of torn diagrams.

There is much more detail than that in Apple's Reference Manual. RAM read/write cycles are shown in relation to master timing, plus use of Direct Memory Access (DMA) by peripherals. The automatic dynamic RAM refresh system explanation and relationship to video display is excellent, good enough to be an extension course for a pro. Lo-Res and Hi-Res color generation is very complete and includes color phase diagrams. An appendix covers NTSC color TV standards with differences between broadcast and Apple signals.

The author claims the book's main purpose is that of providing reference data for custom peripherals. This is quite true, although 6522 PIA users have to do some hunting for proper clocking. The RDY peripheral circuit connection and DMA (Direct Memory Access) operation is well-detailed for experimenters. These two have been sorely neglected, perhaps due to software orientation of most publications.

Descriptions omit European versions and PAL (Phase Alternating Line) TV characteristics. Disk controller and ROM cards are not covered although both are made by Apple; Apple Computer contributed many illustrations and references for the book. I would suggest that Sams is running a trial of an "advanced" book in the North American market. Too many publishers hold to "simple" books.

After 30 years of electron pushing and 20 years in circuit design, I am *very* skeptical of new publications. This is a darn good piece of work with very few nits to pick. There might be some differences between revision descriptions and actual hardware (such as my revision 6), but these are slight. My only real complaint is that this book wasn't available in 1980 when the old Red Book schematics were analyzed to see if the Apple][met my fussy standards.

Author Gayler and publisher Sams have the right stuff. Hopefully we won't have to wait until the *IIe* is replaced to get a similar work for that model. Or the *III* or the "Macintosh."

The Rh+ Factor for Programmers & Authors

We represent freelance programmers and we are looking for creative new Apple programs and program authors to represent to established industry software publishers.

***No evaluation or consulting fees**
***10% Commission on Royalties (15% First Sale)**
***Please send query letter first**

Robert Holiday

The Holly Agency
(312) 463-7484

4049 W. Peterson Ave.
Chicago, IL 60646

Circle 10 on Reader Service Card

Apple Mechanic's hi-res type routines and fonts are usable in your programs WITHOUT LICENSING FEE. Just give Beagle Bros credit on your disk and documentation.

APPLE MECHANIC

HI-RES SHAPE EDITOR / TYPE FONT DISK
by BERT KERSEY

$29.50: Includes Peeks/Pokes Chart & Tip Book #5.

SHAPE EDITOR: Keyboard-draw hi-res shapes for animation in your Applesoft programs. Access & create **proportionally-spaced hi-res Typefaces** with each character re-definable as you want. Six fonts are included on the disk. Excellent LISTable Applesoft demos show you how to animate graphics and create professional-looking Charts and Graphs.

BYTE-ZAP: Rewrite any byte on a disk for repair or alteration. Load entire sectors on the screen for inspection. **Hex/Dec/Ascii displays** and input. Educational experiments included for making trick file names, restoring deleted files, changing DOS, etc.

MORE: Useful music, text and hi-res tricks for your programs. Clear educational documentation.

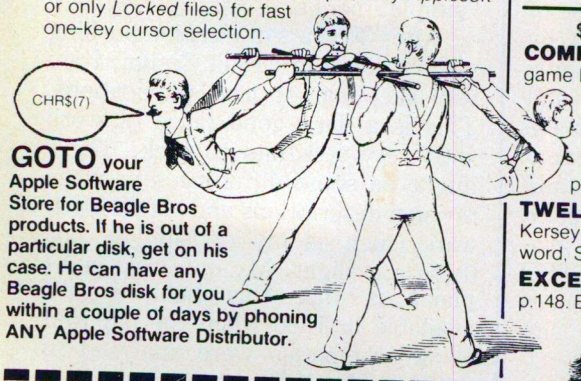
APPLE MECHANIC TYPEFACES

by BERT KERSEY

$20.00: Includes Peeks & Pokes Chart.

26 NEW FONTS for use with Apple Mechanic programs. Many different sizes and typestyles, both ordinary and *Artistic*. Every character—from A to Z to "★" to "□"—of every typeface—from "Ace" to "Zooloo"—is re-definable to suit your needs. All typefaces are **proportionally spaced** for a more professional appearance. People *do* notice the difference!

BEAGLE MENU: Display only the file names you want from your disks (for example, only *Applesoft* or only *Locked* files) for fast one-key cursor selection.



GOTO your Apple Software Store for Beagle Bros products. If he is out of a particular disk, get on his case. He can have any Beagle Bros disk for you within a couple of days by phoning ANY Apple Software Distributor.

RUSH the following disks by First Class Mail—

☐ Alpha Plot	$39.50	☐ Frame-Up	$29.50
☐ Apple Mechanic	29.50	☐ GPLE	49.95
☐ A.M.Typefaces	20.00	☐ ProntoDOS	29.50
☐ Beagle Bag	29.50	☐ Silicon Salad	24.95
☐ Beagle BASIC	34.95	☐ Tip Disk #1	20.00
☐ DiskQuik	29.50	☐ Utility City	29.50
☐ DOS Boss	24.00	☐	
☐ Double-Take	34.95	☐ ADD ME to mailing list.	
☐ Flex Type	29.50	☐ ALREADY ON mail list.	

AT YOUR APPLE DEALER NOW!
Or order directly from Beagle Bros—

VISA — mastercharge

Visa/MasterCard or COD, call **TOLL-FREE**

Orders only / ALL 50 STATES / 24 Hours a Day

1-800-227-3800 ext. 1607

OR mail U.S.Check, Money-Order or Visa/MC# to **BEAGLE BROS**, 8th Floor, 4315 SIERRA VISTA, SAN DIEGO, CA 92103

Add $1.50 First Class Shipping. Any-Size Order. Overseas add $4.00. COD add $3.00. California add 6%.
ALL ORDERS SHIPPED IMMEDIATELY.

AD#8A

DOS BOSS

DISK COMMAND EDITOR
by BERT KERSEY and JACK CASSIDY

$24.00: Includes Peeks/Pokes Chart & Tip Book #2.

RENAME DOS COMMANDS & Error Messages—"Catalog" can be "Cat"; "Syntax Error" can be "Oops" or almost *anything* you want it to be.

PROTECT YOUR PROGRAMS. An unauthorized Save-attempt can produce a "Not Copyable" message, or *any* message you want. Also easy List-Prevention and other useful Apple tips and tricks. Plus one-key program-execution from catalog.

CUSTOMIZE DOS. Change the catalog Disk Volume heading to your message or title. Omit or alter catalog file codes. Fascinating documentation, tips and educational Apple experiments.

ANYONE USING YOUR DISKS (booted or not) will be using DOS the way YOU designed it.

```
10 LIST: LIST: LIST: FOR ZZ=PEEK(175)+PEEK
(176)*256+36 TO 3072: POKE ZZ,216: NEXT
20 FOR XXX=1 TO 2: POKE-16299,0: POKE
-16300,0: XXX=1: NEXT: REM Experiment
with different length variable names.
```

BEAGLE BAG

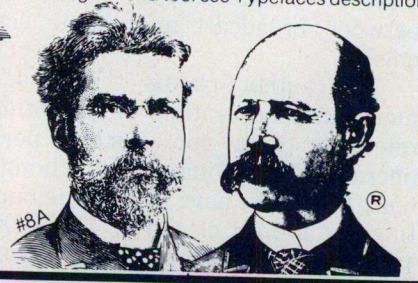
12 APPLE GAMES ON ONE DISK
by BERT KERSEY

$29.50: Includes Peeks & Pokes Chart

COMPARE BEAGLE BAG with any single-game Locked-Up disk on the market today. All 12 games are a blast, the price is a bargain, the instructions are crystal clear, *and* the disk is COPYABLE. You can even *change* the programs or list them to learn programming tricks by seeing how they work.

TWELVE GAMES from the Applesoft Ace, Bert Kersey— TextTrain, Wowzo, Magic Pack, Buzzword, Slippery Digits, and many many more...

EXCELLENT REVIEWS—See Jan-83 *Softalk*, p.148. Beagle Menu too: see Typefaces description.



Beagle Bros Micro Software Inc.

4315 SIERRA VISTA / SAN DIEGO, CA 92103
619-296-6400

ALL BEAGLE DISKS ARE UNLOCKED, COPYABLE AND COMPATIBLE WITH APPLE II, II+ AND IIe.* (Don't Settle for Less!)

*DISKQUIK requires Apple IIe.
"APPLE" is a Registered Trade Mark of You-Know-Who.

NEW! SILICON SALAD

INCLUDING TIP DISK #2
by BERT KERSEY and MARK SIMONSEN

$24.95: Includes Peeks/Pokes AND Commands Charts

MANY MINI-UTILITIES: Disk Scanner finds bad disk sectors, **Key-Clicker** adds subtle sound as you type, **DOS-Killer** adds two tracks of space to your disks, **2-Track Cat** allows up to 210 file names per disk, **Program Splitter** makes room for hi-res pix with large Applesoft programs, **Text Imprinter** transfers text to the hi-res screen, **Onerr Tell Me** prints the appropriate error message but continues program execution, **Text Screen Formatter** converts text layouts into Print statements... plus much more Apple wizardry from the boys at Beagle Bros.

MORE TIPS ON DISK: Including fantastic programming tricks from Beagle Bros Tip Books 5, 6 and 7, plus programs from Tips/Tricks Chart #1.

TWO-LINERS TOO: From our customers around the world—and elsewhere. Little mind-blowers that will teach your old Apple some new tricks!

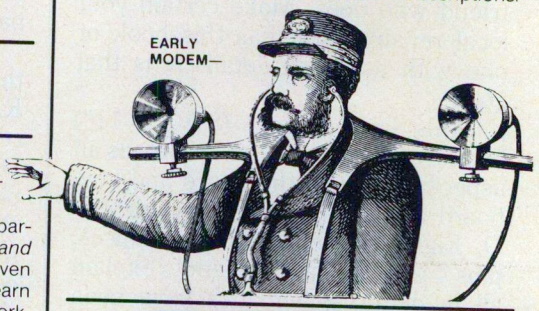
TIP DISK #1

100 TIP BOOK TIPS ON DISK
by BERT KERSEY

$20.00: Includes Peeks & Pokes Chart.

100 LISTABLE PROGRAMS from Beagle Bros Tip Books 1-4. Make your Apple do things it's never done! All 100 programs are LISTable and changeable for Apple experimentation.

COMMAND CHART INCLUDED: Free with each Tip Disk; an 11 x 17 poster of all Applesoft, Integer Basic & DOS Commands with Descriptions!



FLEX TYPE

(FORMERLY "FLEX TEXT")
VARIABLE-WIDTH HI-RES TEXT UTILITY
by MARK SIMONSEN

$29.50: Includes Peeks & Pokes Chart

PRINT VARIABLE-WIDTH TEXT on both hi-res screens with normal Applesoft commands (including HTAB 1-70). Normal, expanded & compressed text with no extra hardware. (70-column text requires a monochrome monitor, not a tv).

ADD GRAPHICS TO TEXT or add Text to hi-res graphics. Run your existing Applesoft programs under Flex Type control. Fast, easy to use, and Compatible with GPLE and Double-Take.

DOS TOOL KIT® font compatibility, or use the supplied Flex Type typefaces. Select up to 9 fonts with control-key commands. A text character editor lets you redesign any Apple text character.

FRAME-UP

FAST APPLE DISPLAY UTILITY
by TOM WEISHAAR

$29.50: Includes Peeks & Pokes Chart

PROFESSIONAL PRESENTATIONS: Turn your existing Hi-Res, Lo-Res and Text frames into attractive Apple "slide shows". *FAST* hi-res loads in 2½-seconds! Paddle or Keyboard-advance frames.

UNATTENDED SHOWS are optional, with each picture arranged and pre-programmed to display on the screen from 1 to 99 seconds. Custom **Text Screen Editor** lets you create black-and-white text "slides" and add type "live" from the keyboard during shows. Mail copies of presentations on disk to your friends and associates (or home to Mom!).

NEW! GPLE

GLOBAL PROGRAM LINE EDITOR
by NEIL KONZEN

$49.95: Includes Peeks/Pokes Chart & Tip Book #7.

A CLASSIC APPLE PROGRAM EDITOR GPLE lets you edit Applesoft program lines *FAST* without awkward cursor-tracing and "escape editing".

INSERT & DELETE: GPLE works like a word processor for Applesoft program lines. You make changes instantly by jumping the cursor to the change point and inserting or deleting text. No need to trace to the end of a line before hitting Return.

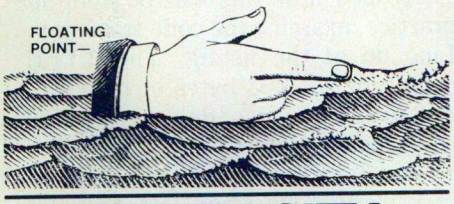
GLOBAL SEARCH & REPLACE: Find any word or variable in your programs, *FAST*. For example, find all lines containing a GOSUB, or edit or delete all lines with REM statements, or all occurrences of any variable. **Replace any variable**, word or character with any other. For example, change all X's to ABC's, or all "Horse" strings to "Cow".

80-COLUMN COMPATIBILITY: All edit & global features support **Apple IIe 80-column cards** and most 80-column cards on *any* Apple IIe, II+ or II.

DEFINABLE ESC FUNCTIONS: Define ESC plus any key to perform any task. For example, **ESC-1** can catalog drive 1, **ESC-L** can do a "HOME: LIST", **ESC-N** could type an entire subroutine... *Anything* you want, *whenever* you want.

GPLE DOS MOVER: Move DOS and GPLE to Language Card (or IIe upper 16K) for an EXTRA **10,000 Bytes** (10K) of programmable memory.

Plus APPLE TIP BOOK #7: Learn more about your Apple! Includes all new GPLE tips and tricks.



UTILITY CITY

21 PROGRAMMING UTILITIES
by BERT KERSEY

$29.50: Includes Peeks/Pokes Chart & Tip Book #3

LIST FORMATTER prints each program statement on a new line. Loops indented with printer Page Breaks. A great Applesoft program de-bugger.

MULTI-COLUMN CATALOGS, with or without sector and file codes. Organize your disk library.

INVISIBLE and trick catalog file names. Invisible functioning commands in Applesoft programs too.

MUCH MORE: 21 utilities, including auto-post Run-number & Date in programs, alphabetize/store info on disk, convert dec to hex or Int to FP, protect and append programs, dump text to printer...

LEARN PROGRAMMING: List-able programs and informative documentation. Includes Tip Book #3. Hours of good reading & Applesoft experiments.

ALPHA PLOT

HI-RES GRAPHICS/TEXT UTILITY
by BERT KERSEY and JACK CASSIDY

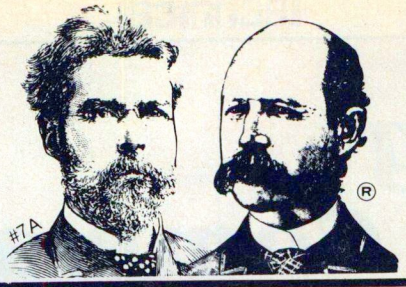
$39.50: Includes Peeks/Pokes Chart & Tip Book #4.

DRAW IN HI-RES on both Apple "pages" using easy keyboard commands OR paddles/joystick. Pre-view lines before plotting. Solid or mixed colors & Reverse (background-opposite) drawing. FAST one-keystroke circles, boxes & ellipses, filled or outlined. Add text for graphs & charts. All pix Save-able to disk, to be called from your Applesoft programs.

COMPRESS HI-RES DATA to 1/3 disk space (average) allowing more hi-res pictures per disk.

MANIPULATE IMAGES: Superimpose any two images, or RE-LOCATE any rectangular section of any drawing anywhere on either hi-res page.

HI-RES TYPE: Add text to your pictures with adjustable character-size and large-character color. Type *anywhere* with no Htab/Vtab limits. Type sideways too, for graphs. Includes Tip Book #4.



Beagle Bros Micro Software Inc.

4315 SIERRA VISTA / SAN DIEGO, CA 92103
619-296-6400

ALL BEAGLE DISKS ARE UNLOCKED, COPYABLE AND COMPATIBLE WITH APPLE II, II+ AND IIe.*

(Don't Settle for Less!)

* DISKQUIK requires Apple IIe.
"APPLE" is a Registered Trade Mark of You-Know-Who.

BEAGLE BASIC

APPLESOFT ENHANCER
by MARK SIMONSEN

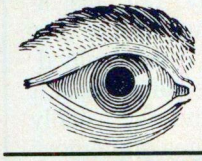
$34.95: Includes Peeks/Pokes Chart & Tip Book #6. Requires Apple IIe (OR II/II+ with RAM Card).

RENAME ANY APPLESOFT COMMAND or Error Message to anything you want. For program clarification, encryption/protection or even foreign translation. Plus add optional NEW COMMANDS:

ELSE follows If-Then statements, like this:
IF X=2 THEN PRINT "YES": *ELSE* PRINT "NO"

HSCRN reads color of any hi-res dot for collision testing. **SWAP X,Y** exchanges 2 variables' values. New **TONE** command writes music with no messy pokes & calls. **SCRL** scrolls text in *either* direction. **TXT2** lets Text Page 2 act exactly like Page 1.

PLUS: GOTO & GOSUB may precede variables, as in "GOSUB FIX" or "GOTO 4+X". Escape-mode indicated by special ESC CURSOR. Replace awkward Graphics screen-switch pokes with 1-word commands. Change ctrl-G Beep to any tone. **INVERSE REMS** too! All GPLE compatible.



```
1 FOR S=768 TO 773: READ A:
  POKE S,A: NEXT: POKE 232,0:
  POKE 233,3: DATA 1,0,4,0,5,0
2 HGR2: FOR R=0 TO 192: ROT=R:
  SCALE=96: XDRAW 1 AT 140,95:
  SCALE=30: XDRAW 1 AT 140,95:
  S=PEEK(49200): NEXT: RUN
```

PRONTO-DOS

HIGH-SPEED DOS / DOS-MOVE UTILITY
by TOM WEISHAAR

$29.50: Includes Peeks & Pokes Chart

TRIPLES THE SPEED of disk access and frees 10,000 bytes of extra memory by moving DOS.

Function	Normal	Pronto
BLOAD HI-RES IMAGE	10 sec.	3 sec.
LOAD 60-SECTOR PROGRAM	16 sec.	4 sec.
SAVE 60-SECTOR PROGRAM	24 sec.	9 sec.
BLOAD LANGUAGE CARD	13 sec.	4 sec.
(Text Files: No Change)		

Boot the Pronto disk or your updated disks, created with the normal INIT command. Compatible with all DOS Commands, GPLE, Double-Take, DOS Boss, DiskQuik and almost all unprotected programs.

MOVE DOS to your Language Card, RAM Card, or standard Apple IIe upper 16K, freeing up **10,000 EXTRA BYTES** of memory for your programs.

15 EXTRA SECTORS per disk. Catalog Free-Space displayed every time you catalog a disk.

TYPE-COMMAND ("TYPE filename") prints contents of sequential Text Files on screen or printer.

NEW! DISKQUIK

DISK DRIVE EMULATOR
by HARRY BRUCE and GENE HITE

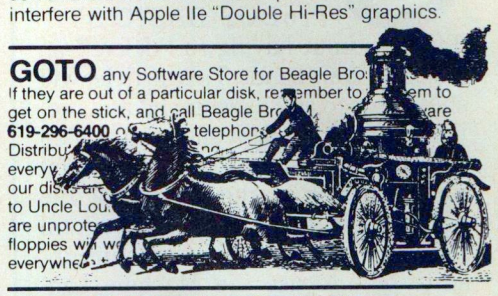
$29.50: Includes Peeks & Pokes Chart
Requires Apple IIe with Extended 80-column Card.

ACTS LIKE A DISK DRIVE in Slot 3, but much faster, quieter, more reliable and $350+ cheaper! Enjoy the benefits of a 2nd (or 3rd or 4th...) drive at less than 1/10th the price. Catalogs normally with "CATALOG, S3" command. Load & Save any kind of files into RAM with normal DOS commands.

SILENT AND FAST: Since no moving parts are involved, DiskQuik operates silently and at super-high speeds. See it to believe it. Your Apple IIe's Extended 80-column Card (required) can hold about half the amount of data as a 5¼" floppy disk!

MANY USES: For example, auto-load often-used files like FID etc., etc., into RAM when you boot up, so they are always available when you need them. Copy files from RAM onto disk and vice versa, just as if a disk drive were connected to slot #3.

FRIENDLY & COMPATIBLE with 80-column display, GPLE, ProntoDOS, and all normal Applesoft and DOS commands and procedures. Will not interfere with Apple IIe "Double Hi-Res" graphics.



DOUBLE-TAKE

2-WAY-SCROLL/MULTIPLE UTILITY
by MARK SIMONSEN

$34.95: Includes Peeks/Pokes AND Tips/Tricks Charts.

2-WAY SCROLLING: Listings & Catalogs scroll Up AND Down, making file names and program lines much easier to access. Change the Catalog or List scroll-direction at will, with Apple's Arrow keys.

80-COLUMN COMPATIBLE: All features support IIe and most other 80-column cards.

BETTER LIST FORMAT: Each program statement lists on a new line for *FAST* program tracing & de-bugging. Printer-compatible; any column-width.

VARIABLE-DISPLAY: Displays all of a program's strings and variables with current values.

CROSS-REFERENCE: Sorts and displays line numbers where each variable & string appears.

AUTO-LINE-NUMBER, Hex/Dec Converter, better Renumber/Append, Program Stats, Change Cursor, Space-On-Disk. GPLE/Pronto compatible.

☐ Alpha Plot	$39.50	☐ Frame-Up	$29.50
☐ Apple Mechanic	29.50	☐ GPLE	49.95
☐ A.M.Typefaces	20.00	☐ ProntoDOS	29.50
☐ Beagle Bag	29.50	☐ Silicon Salad	24.95
☐ Beagle BASIC	34.95	☐ Tip Disk #1	20.00
☐ DiskQuik	29.50	☐ Utility City	29.50
☐ DOS Boss	24.00	☐	
☐ Double-Take	34.95	☐ ADD ME to mailing list.	
☐ Flex Type	29.50	☐ ALREADY ON mail list.	

AT YOUR APPLE DEALER NOW!
Or order directly from Beagle Bros—

VISA · master charge

Visa/MasterCard or COD, **call TOLL-FREE**
Orders only / ALL 50 STATES / 24 Hours a Day

1-800-227-3800 ext. 1607

OR mail U.S.Check, Money-Order or Visa/MC#
to **BEAGLE BROS**, 7th Floor
4315 SIERRA VISTA, SAN DIEGO, CA 92103

Add $1.50 First Class Shipping. Any-Size Order
Overseas add $4.00 COD add $3.00 California add 6%
ALL ORDERS SHIPPED IMMEDIATELY.

CLIP COUPON OR USE SEPARATE SHEET

Adding a TYPE Command to ProDOS

Cecil Fretwell

ONE of the many powers of ProDOS is the ability to add a command to the ProDOS/BASIC.SYSTEM without having to "delete" an existing command. This article will show how to add a command and provides an example command called TYPE whose purpose is to list a text file from disk to the screen (or the printer if it is active). The program for TYPE also provides some insight as to how to use the Machine Language Interface (MLI).

The command is simple to use. Merely enter the command in the following form.

TYPE pathname

If the pathname is omitted, the command aborts with SYNTAX ERROR. If the specified file does not exist on disk, the command aborts with the PATH NOT FOUND error. If the file specified by pathname is not a text file (TXT), the command aborts with the FILE TYPE MISMATCH message. The command may also abort with an appropriate message as a result of an MLI operation.

A successful use of TYPE will result in the display of the specified text file. This is performed without disturbing any existing program in the system, i.e., the file is read from disk and sent directly to the screen. In the listing process, entry of [Ctrl-C] causes the command to abort in an orderly manner. [Ctrl-S] may be used to halt the display process which may then be aborted with [Ctrl-C] or continued by entering anything other than [Ctrl-S] or [Ctrl-C].

The TYPE command may also be invoked using the CHR$(4) concept in a BASIC PRINT statement. Care should be taken to insure that no other files are active while the TYPE command is active.

This article will provide a general description of how to add a command. The listing of TYPE will not be discussed in detail; its purpose is to provide a reference for the hacker or to provide code for one who merely wishes to install the TYPE capability. In the discussion to follow, the use of the words system, DOS, or ProDOS all refer to the combination of the kernel called the Machine Language Interface and BASIC.SYSTEM which taken together one could relate to the old DOS 3.3.

For the user who wants to install TYPE without wading through the discussion to follow, please refer to the INSTALLATION OF TYPE section given at the end of this article.

The most logical place to insert a user command is immediately below BASIC.SYSTEM and other existing user commands, if any. For this insertion process, the length of the user command code must be determined. Except for very simple commands, it appears that a command will usually require at least two pages of memory unless one is shrewd in their programming efforts.

Given the starting address of the beginning of all user command codes or the beginning of BASIC.SYSTEM ($9A00), if no commands are installed, subtract the new command code length from that address. Now retain only the first two hex digits in the result, subtract 4 from that value, and store the final result in $74. This results in a HIMEM value which satisfies the rule that HIMEM must point to a page boundary and must be at least 1K below all system code.

The business of moving HIMEM down has a potential hidden "gotcha." Existing Applesoft strings will be mangled unless special precautions are taken. How this can be avoided will soon be described.

Now the new command code can be moved to its final destination 1K or more above HIMEM, then protected via the system bit map. Finally, the system must be "modified" to recognize the new command.

To see how this can be done in practice, examine the code starting in Line 66 of the listing. The TYPE code requires 203 bytes of memory. A convenient way of determining this length and providing a most efficient way of testing the code at the same time without adjusting HIMEM will be shown in a future paragraph.

Given a length of 203 bytes, HIMEM must be moved down one page. Lines 66 and 67 load the A register with the number of pages required, 1, then call GETBUFR ($BEF5). This call not only moves HIMEM down one page, but also insures that string storage has been preserved, i.e., it has been moved down in memory. If this process fails, an error message is displayed and the installation process is immediately aborted. Current documentation does not indicate what type of error may occur so it would probably be a safe precaution to immediately re-boot the system.

Lines 86 to 103 move the TYPE code from the $2000 area to its final destination 1K above the new HIMEM. Even though TYPE requires less than one page, the code in Lines 86 to 103 provides a general scheme of moving multiple pages. The code has been designed to be almost position independent. Suppose HIMEM was $9300 before the installation of TYPE. After the installation, HIMEM will be $9200 and TYPE's code will reside at $9600. If

```
                        ****************************************
                        * TYPE command for PRODOS BASIC.SYSTEM *
                        *          By  Cecil Fretwell          *
                        *           Call -A.P.P.L.E.           *
                        ****************************************

                        ****************************************
                        * Prepared using Big Mac LC assembler  *
                        ****************************************

                        ************************************************************
                        * This command reads a text file from disk and displays    *
                        * its contents on the screen.                              *
                        *                                                          *
                        * To invoke the command, enter                             *
                        *      ]TYPE pathname                                      *
                        * If the pathname is not given, the command will abort     *
                        * with the PATH NOT FOUND error.  If the file does not     *
                        * exist, the command will abort with the PATH NOT FOUND    *
                        * Error.  If the specified file is not a text file, the    *
                        * Command aborts with the FILE MIS MATCH error.            *
                        *                                                          *
                        * The display process may be aborted by means of           *
                        * [Ctrl-C].  [Ctrl-S] may be used to halt the display      *
                        * Process.  Entry of other than [Ctrl-S] or [Ctrl-C]       *
                        * continues the display process.                           *
                        *                                                          *
                        * If any other TYPE of MLI error occurs, the command       *
                        * aborts after displaying an appropriate error code        *
                        * based upon the gosystem MLI error code.                  *
                        ************************************************************

                HIMEM    =    $73          ;Himem pointer
                INBUF    =    $200         ;Keyboard input buffer
                EXTRNCMD =    $BE06        ;External cmd jmp vector
                XTRNADDR =    $BE50        ;External command impl address
                XLEN     =    $BE52        ;Length of command string-1
                XCNUM    =    $BE53        ;CLI cmd no. (ex cmd = 0)
                PBITS    =    $BE54        ;Command parameter bits
                FBITS    =    $BE56        ;Command parameter bits found
                GOSYSTEM =    $BE70        ;Perform MLI call for code in a
                SSGINFO  =    $BEB4        ;Get file info MLI buffer
                FIFILID  =    $BEB8        ;File id of disk file
                OSYSBUF  =    $BECE        ;Buffer for MLI open
                OREFNUM  =    $BED0        ;Reference number of opened file
                NEWLREF  =    $BED2        ;New line file reference number
                NLINENBL =    $BED3        ;New line enable mask and char
                RWREFNUM =    $BED6        ;Read file reference number
                RWDATA   =    $BED7        ;Read buffer = inbuf
                RWCOUNT  =    $BED9        ;Read-number of char requested
                RWTRANS  =    $BEDB        ;Read-number of char transferred
                CFREFNUM =    $BEDE        ;Close file reference number
                GETBUFR  =    $BEF5        ;Get buffer from system
                BITMAP   =    $BF58        ;System bit map table
                KBD      =    $C000        ;Keyboard input port
                KBDSTRB  =    $C010        ;Keyboard strobe
                PRBYTE   =    $FDDA        ;Display byte in a in hex
                COUT     =    $FDED        ;Display A on screen

                         ORG  $1F61

                         *******************************************
                         * Get space for TYPE code and move string *
                         * storage down, if necessary.             *
                         *******************************************
1F61: A9 01              LDA  #1           ;Need 1 page
1F63: 20 F5 BE           JSR  GETBUFR      ;Go get it
1F66: 90 2E              BCC  LOADER       ;If success
1F68: 48                 PHA               ;Failed, save error code
1F69: A2 00              LDX  #0           ;Display message
```

```
1F6B: BD 7A 1F   71  ERTYPE   LDA  ERTYPEM,X
1F6E: F0 06      72           BEQ  ERTYPN      ;If end of message
1F70: 20 ED FD   73           JSR  COUT
1F73: E8         74           INX
1F74: D0 F5      75           BNE  ERTYPE      ;Always branch
1F76: 68         76  ERTYPN   PLA              ;Recover error code
1F77: 4C DA FD   77           JMP  PRBYTE      ;Display and return to system
1F7A: 87 87      78  ERTYPEM  DFB  $87,$87
1F7C: D4 D9 D0   79           ASC  "TYPE NOT INSTALLED ERROR "
1F7F: C5 A0 CE CF D4 A0 C9 CE
1F87: D3 D4 C1 CC CC C5 C4 A0
1F8F: C5 D2 D2 CF D2 A0
1F95: 00         80           DFB  0

                 82  ****************************************
                 83  * Move the TYPE code to its true home *
                 84  ****************************************

1F96: 8D AD 1F   86  LOADER   STA  LOADM+2     ;Plant base address
1F99: 8D D3 1F   87           STA  NXTP1+2
1F9C: 8D D9 1F   88           STA  NXTP2+2
1F9F: 8D FD 1F   89           STA  BASEA
1FA2: 8D 0B 20   90           STA  BASE+TYPECS-TYPEC+2
1FA5: 8D 26 20   91           STA  BASE+TYPEDO-TYPEC+1
1FA8: AD 00 20   92  LOAD     LDA  BASE        ;Get byte of code
1FAB: 8D 00 20   93  LOADM    STA  TYPEC       ;Move it
1FAE: EE A9 1F   94           INC  LOAD+1      ;Advance addresses
1FB1: D0 03      95           BNE  LOAD1
1FB3: EE AA 1F   96           INC  LOAD+2
1FB6: EE AC 1F   97  LOAD1    INC  LOADM+1
1FB9: D0 03      98           BNE  LOAD2
1FBB: EE AD 1F   99           INC  LOADM+2
1FBE: CE FE 1F   100 LOAD2    DEC  LENGTH      ;Done with move?
1FC1: D0 E5      101          BNE  LOAD        ;If definitely not
1FC3: CE FF 1F   102          DEC  LENGTH+1    ;Done with move?
1FC6: 10 E0      103          BPL  LOAD        ;If definitely not

                 105 ****************************************
                 106 * Protect TYPE code via system bit map *
                 107 ****************************************

1FC8: AD FD 1F   109          LDA  BASEA       ;Get page address for page code
1FCB: 20 E6 1F   110          JSR  BITMAPS     ;and update bit map

                 112 ********************************************************
                 113 * Finish up by installing the command.  This is       *
                 114 * first time processing to install the TYPE command *
                 115 ********************************************************

1FCE: AD 07 BE   117          LDA  EXTRNCMD+1  ;Move current command handler
1FD1: 8D 0F 20   118 NXTP1    STA  NXTCMD+1    ;to point to next command, if any
1FD4: AD 08 BE   119          LDA  EXTRNCMD+2
1FD7: 8D 10 20   120 NXTP2    STA  NXTCMD+2
1FDA: A9 04      121          LDA  #<TYPE      ;Install the address of our
1FDC: 8D 07 BE   122          STA  EXTRNCMD+1  ;command handler in the
1FDF: AD FD 1F   123          LDA  BASEA       ;External command
1FE2: 8D 08 BE   124          STA  EXTRNCMD+2  ;Jmp vector
1FE5: 60         125          RTS              ;Return to caller

                 127 ************************************
                 128 * Subroutine to update bit map.    *
                 129 * Enter with A = page number to    *
                 130 * mark in the bit map.             *
                 131 ************************************

1FE6: AA         133 BITMAPS   TAX              ;Save page number in x
1FE7: 4A         134           LSR              ;Compute offset into bit map
1FE8: 4A         135           LSR
1FE9: 4A         136           LSR
1FEA: A8         137           TAY              ;Y = offset
1FEB: 8A         138           TXA              ;A = page number
1FEC: 29 07      139           AND  #7          ;Isolate bit number
```

HIMEM was $9600 before installation then HIMEM will be $9500 after installation. Lines 86 to 91 provide some addresses consistent with the final home of TYPE.

Note that TYPE is stored at the beginning of a page boundary yet is less than one page long. Granted this wastes some space at the end of TYPE, however starting on a page boundary simplifies the relocation process. Even if TYPE nestled up against the next user code or BASIC.SYSTEM, HIMEM can not be moved up to remove this wasted space. *HIMEM must always contain a value that represents a page boundary.*

Lines 109 and 110 protect the page required for TYPE. This code makes use of the subroutine in Lines 133 to 148. Even though a subroutine is not necessary, it does provide a general scheme for protecting multiple pages. For example, if TYPE required two pages, incrementing BASEA then calling BITMAPS protects the second page. This is very important because it appears that if one fails to protect the command code, any future operation involving disk operations via a system command will wipe out some or all of the user command code.

Lines 117 to 125 tell the system about the new command via EXTRNCMD at $BE06 in BASIC.SYSTEM's Global Page. This logic is perfectly general in that it picks up the current value of EXTRNCMD and stuffs it into the TYPE logic. The end result is that if the command is not TYPE, a link is made to the next user command logic or a return is made to the system. More on this later. Having moved the current EXTRNCMD value, EXTRNCMD is set to point to the beginning of the TYPE logic.

The TYPE command is now installed. Before proceeding to its logic, a note on a convenient method of testing a new command for correct operation and determining its final length. In the test stage, Lines 66 to 110 did not exist. BRUNing this "modified" version of the command resulted in its residing in a "safe" area of memory with the TYPE command active. Obviously the integrity of a loaded BASIC program can not be guaranteed. Based on experience, this saves a lot of headaches. Once the command was fully tested, Lines 66 to 110 were inserted to produce the final code. The strange origin of $1F61 was chosen to provide a TYPE listing beginning on a page boundary.

In order to understand the command process, when a command is entered, the system first checks for a standard DOS command such as RUN, BLOAD, etc. If a standard command is found, the system proceeds accordingly. If a standard DOS command is not found, the system performs a JSR to EXTRNCMD. If a user command exists, it performs its thing then returns the carry flag clear which causes the command process to exit cleanly. If a user command does not exist or the command does not belong to the user, the carry flag is returned set and the system passes the command to Applesoft. If the command is an Applesoft command, that command is performed then the command process is exited. If the command can not be processed by one of these three possibilities, the pesky SYNTAX ERROR message will appear.

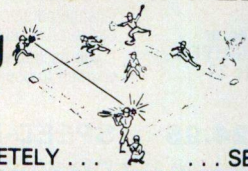
Assuming a user command exists, the first piece of code executed after the JSR EXTRNCMD must check the keyboard input buffer at $200 for the first user command installed. If the command is not the first user command, the logic must jump to the next user command installed or return the carry flag set. The next user command logic looks for its command in the keyboard buffer. If the command is not found, the logic must jump to the next user command installed or return the carry flag set. And so on through the user chain of commands.

If the command belongs to the user, it can be processed in two stages. In the first stage, besides any special application dependent processing, the user is required to establish certain conditions before returning to the system. At some place in the first stage code, the contents of XCNUM ($BE53) must be set to zero to mark the command as external to the system. This also requires that the length of the command minus one be placed in XLEN ($BE52).

Second, even if the user command does not require second stage processing, an address must be stored in XTRNADDR ($BE50). This address specifies the location for the second

~~TRIPLE~~ DOUBLE DENSITY. . . ON STANDARD APPLE DRIVES?

Introducing TRIPLE PLAY



A COMPLETELY . . . . . SELF CONTAINED ENHANCEMENT OF APPLESOFT THAT FEATURES:

* *NUMERIC DATA COMPRESSION STORE 3½ TIMES MORE INTEGERS! STORES 3 TIMES MORE REALS!*
* *TEXT AND NUMBERS CAN BE INTERMIXED!*
* *SYNTAX CHECKING WORKS EXACTLY LIKE **PRINT** AND **INPUT** STATEMENTS!*
* *USE WITH SEQUENTIAL OR RANDOM FILES!*

USE TECHNIQUE WITH ANY DOS OR DISK SYSTEM (FLOPPY, HARD OR RAM)

AND IT'S FASTER THAN NORMAL I/O ROUTINES!

IF YOU STORE NUMBERS THEN YOU NEED TRIPLE PLAY. GET AT LEAST 3 TIMES MORE NUMBERS PER DISKETTE.

UNPROTECTED PROGRAM DISK WITH COMPLETE DOCUMENTATION ONLY . . . $19.95

TO ORDER: Ask for "TRIPLE PLAY"

send check or money order to:

GRAYSOFT
P.O. Box 515
Buffalo, WY 82834

307-684-7257

PLEASE ALLOW 2 WEEKS FOR DELIVERY. FOREIGN ORDERS MAY REQUIRE MORE POSTAGE.

(NOTE: APPLE & APPLESOFT ARE REGISTERED TRADEMARKS OF APPLE COMPUTERS INC.)

Circle 20 on Reader Service Card

stage processing. If the user has no second stage processing, XTRNADDR must point to a CLC RTS sequence or a simple RTS if it is known that the carry flag is cleared. If second stage processing is required, XTRNADDR must point to that code. This is most important if the user has more than one command installed.

Third, the user must establish the desired parameter bits in PBITS ($BE54,$BE55). The standard parameters A, B, E, L, S, D, F, R, V, @, and T can be used to one's advantage because before second stage processing is invoked, the system will parse the parameters, check for errors, etc. and return valuable information to second stage processing. For example, in the TYPE command, a pathname to the desired file must be entered. One could check for a pathname in the first stage processing but why waste code or such things as reporting an error if a name is not given. Instead PBITS is set to 1 and PBITS+1 is set to 4. The significance of this will soon become clear.

If the user does not use parameters then all of PBITS must be set to zero. If standard parameters are used, the manuals provide well documented information regarding the bits to be used in PBITS. In this process, the user can not mix their own parameter symbols with the standard symbols. If such is the case, the user is responsible for parsing the entire line with an end result of zeroing PBITS. This can be non-trivial code, therefore, why not let the system do all the work at the cost of a few lines of documentation describing the meaning of a standard symbol in the command line.

Another fact to note is that, in general, it can not be assumed that the command line is preserved from first stage processing to second stage processing. As an example, the second stage of TYPE code makes use of the fact that $280 will contain the length of the pathname followed by the pathname itself. Other testing has shown that the beginning of the $200 area is used for PREFIX processing. Apple documents the fact that $200 is used but does not indicate how much of that area is used.

Having performed all of the necessary operations for first stage processing, the user must insure that the carry flag is clear just prior to an RTS to the system. Because of how the system handles the carry flag upon return from the first stage, if error processing is performed in the first stage with the purpose of never invoking the second stage, the user must point XTRNADDR to a CLC RTS sequence then return the carry flag clear. Returning the carry flag set would make the system think the entered command must be passed to Applesoft.

Upon return from the first stage processing, if the system can find nothing wrong with the command and its syntax, control is passed to the second stage processing through XTRNADDR. If one sets bits in the parameter bytes at PBITS during first stage processing, information related to those bits is returned in FBITS ($BE56,$BE57). In the TYPE code, if Bit 0 of FBITS is set, the system found a pathname and processing continues accordingly. If Bit 0 of FBITS is clear, no pathname was entered, an error message must be generated, and a return with the carry flag clear must be made to the system.

Exactly what the system does with PBITS to produce FBITS is not fully explained in the current documentation. Based on studying the code that is executed between first and second stage processing, there is a lot of power in this code. Such will be the topic of many future articles using ProDOS.

Error processing can be quite simple. In TYPE, when no pathname is given, the appropriate message to display is SYNTAX ERROR. This is accomplished by loading the A register with 6 then returning to the system with carry set. The system then produces the SYNTAX ERROR message.

The BASIC Programming with ProDOS manual shows the possible errors handled by the contents of the A register. For those familiar with the ONERR feature of BASIC, error codes 2 through 21 are almost exactly identical in meaning as was the case under DOS 3.3 control.

Other than standard error processing must be provided by user code. Carry must be returned clear after such error processing, other-

ONE PASS COPY **$29.95**

The Copy Machine

Does for disks what Xerox did for paper.

- Don't let backing-up slow you down.
- Examples: Copy Apple System Master on one drive in one pass and 38 seconds compared to two and a half minutes with COPYA.
- **Copy a disk with a 128 sector game on two drives in EIGHTEEN SECONDS (!) vs. 1 minute 33 seconds with COPYA.**
- This is the fastest possible copy system on Apples.
- Make backing up a pleasure. Copy most disks in one pass and a fraction of the time.
- Change parameters — INIT, bypass bad sectors, etc.
- If you backup your work, you owe it to yourself to use ONE PASS COPY.

RAM DRIVE $24.95

- Use your extra memory as a Disk Drive. **No hardware needed.**
- All DOS commands work the same.
- 310 sectors with a 128K Apple IIe, 63 sectors on any 64K Apple IIe or II+. That's more room than any other software.
- Incredibly fast — you have to see it to appreciate the speed and reliability.
- Full package of utilities.

SPEED-DOS $24.95

- Improves SAVE, LOAD, BLOAD, BSAVE, RUN and BRUN times **up to 500%.**
- Compatible with RAM DRIVE, all DOS commands, most programs.
- Bload HI-RES screen from floppy in 3 seconds, from RAM DRIVE in half a second!
- Completely unprotected. Add to any program.

SPECIAL POWER PACK
GET ALL 3 DISKS for only **$49.95**

To Order:

1) Mail or phone orders accepted.
2) Check, COD, VISA or MC (include exp. date and signature)
3) Add $1.50 U.S. shipping
4) Add $5.00 foreign
5) Specify 64K or 128K Apple when ordering RAM DRIVE

- Easy to use documentation.
- All disks copyable, catalogable.
- Dealer inquiries welcome.

Software banc, inc.
1225 N. Water Street
Milwaukee, WI 53202
(414) 271-0100 (312) 876-0715

Circle 26 on Reader Service Card

```
1FEE: AA          140           TAX
1FEF: A9 00       141           LDA  #0          ;Set up mask in A
1FF1: 38          142           SEC
1FF2: 6A          143 BITMAPS1  ROR              ;Shift bit
1FF3: CA          144           DEX
1FF4: 10 FC       145           BPL  BITMAPS1    ;If bit not formed
1FF6: 19 58 BF    146           ORA  BITMAP,Y    ;Set the bit
1FF9: 99 58 BF    147           STA  BITMAP,Y    ;in the bit map
1FFC: 60          148           RTS              ;and return

1FFD: 00          150 BASEA     DFB  0           ;Base page of code

1FFE: CB 00       152 LENGTH    DA   TYPEND-TYPEC ;Length of code to move

                  154 BASE      =    *

                  156 ****************************************
                  157 * The following is the TYPE command code. *
                  158 * This code is moved to its final home by *
                  159 * The code given above.                  *
                  160 ****************************************

                  162 ******************************
                  163 * Our command string is TYPE *
                  164 ******************************

2000: D4 D9 D0    166 TYPEC     ASC  "TYPE" ;This is our command
2003: C5

                  168 ************************************************************
                  169 * Not a ProDOS command so check to see if it is TYPE *
                  170 * command.  If not TYPE command then check for next  *
                  171 * User command or tell ProDOS it is not a user       *
                  172 * Command.  If TYPE command, set up first stage      *
                  173 * processing.                                        *
                  174 ************************************************************

2004: A2 00       176 TYPE      LDX  #0          ;Check for our command
2006: BD 00 02    177 NXTCHR    LDA  INBUF,X     ;Get keyboard character
2009: DD 00 20    178 TYPECS    CMP  TYPEC,X     ;Does it match?
200C: F0 03       179           BEQ  NXTCHRI     ;If yes
200E: 4C 00 00    180 NXTCMD    JMP  0           ;Address planted to next command
2011: E8          181 NXTCHRI   INX              ;Next character
2012: E0 04       182           CPX  #4          ;All characters yet?
2014: D0 F0       183           BNE  NXTCHR      ;No, read next one
2016: A9 00       184           LDA  #0          ;Mark command as external
2018: 8D 53 BE    185           STA  XCNUM
201B: A9 03       186           LDA  #3          ;Put command length-1
201D: 8D 52 BE    187           STA  XLEN        ;In CI global XLEN
2020: A9 36       188           LDA  #<DOTYPE    ;Point to command handler
2022: 8D 50 BE    189           STA  XTRNADDR
2025: A9 20       190 TYPEDO    LDA  #>DOTYPE
2027: 8D 51 BE    191           STA  XTRNADDR+1
202A: A9 01       192           LDA  #1          ;Specify file name expected
202C: 8D 54 BE    193           STA  PBITS
202F: A9 04       194           LDA  #4          ;Fetch prefix, if required
2031: 8D 55 BE    195           STA  PBITS+1
2034: 18          196           CLC              ;Show success on this stage
2035: 60          197           RTS              ;and return to system

                  199 ************************************************
                  200 * Finish TYPE command after ProDOS has given its *
                  201 * blessings to continue.                         *
                  202 ************************************************

2036: AD 56 BE    204 DOTYPE    LDA  FBITS       ;A = found bits
2039: 4A          205           LSR              ;File name found?
203A: B0 04       206           BCS  DOTYPE1     ;If yes
203C: A9 06       207           LDA  #6          ;No, path not found error
203E: 38          208 ERROR     SEC              ;Set carry to show error
203F: 60          209 MLIERR    RTS              ;Return to system with carry set
```

```
2040: A9 0A      210  DOTYPE1  LDA  #$A          ;Set up get file info call
2042: 8D B4 BE   211           STA  SSGINFO
2045: A9 C4      212           LDA  #$C4         ;MLI get file info code
2047: 20 70 BE   213           JSR  GOSYSTEM     ;Perform the MLI call
204A: B0 F3      214           BCS  MLIERR       ;If error, return carry set
204C: AD B8 BE   215           LDA  FIFILID      ;A = file TYPE
204F: C9 04      216           CMP  #4           ;Text file?
2051: F0 04      217           BEQ  DOTYPE2      ;If yes, continue
2053: A9 0D      218           LDA  #13          ;FILE MISMATCH error
2055: D0 E7      219           BNE  ERROR        ;Always branch to return error
2057: A5 74      220  DOTYPE2  LDA  HIMEM+1      ;Set up open
2059: 8D CF BE   221           STA  OSYSBUF+1
205C: A9 C8      222           LDA  #$C8         ;MLI open code
205E: 20 70 BE   223           JSR  GOSYSTEM     ;Perform the MLI call
2061: B0 DC      224           BCS  MLIERR       ;If error, return carry set
2063: AD D0 BE   225           LDA  OREFNUM      ;Set up file reference number
2066: 8D D2 BE   226           STA  NEWLREF      ;For new line call
2069: 8D D6 BE   227           STA  RWREFNUM     ;For read call
206C: 8D DE BE   228           STA  CFREFNUM     ;For close call
206F: A9 7F      229           LDA  #$7F         ;Set up new line call
2071: 8D D3 BE   230           STA  NLINENBL
2074: A9 0D      231           LDA  #$D          ;$D or $8D will terminate line
2076: 8D D4 BE   232           STA  NLINENBL+1
2079: A9 C9      233           LDA  #$C9         ;MLI new line code
207B: 20 70 BE   234           JSR  GOSYSTEM     ;Perform the MLI call
207E: B0 BF      235           BCS  MLIERR       ;If error, return carry set
2080: A9 00      236           LDA  #<INBUF      ;Set up read buffer
2082: 8D D7 BE   237           STA  RWDATA
2085: A9 02      238           LDA  #>INBUF
2087: 8D D8 BE   239           STA  RWDATA+1
208A: A9 00      240           LDA  #0           ;Set up read count
208C: 8D D9 BE   241           STA  RWCOUNT
208F: A9 01      242           LDA  #1
2091: 8D DA BE   243           STA  RWCOUNT+1
2094: A9 CA      244  DOTYPE3  LDA  #$CA         ;MLI read code
2096: 20 70 BE   245           JSR  GOSYSTEM     ;Perform the MLI call
2099: 90 09      246           BCC  DOTYPE5      ;If no error, continue
209B: C9 05      247           CMP  #5           ;End of file?
209D: D0 9F      248           BNE  ERROR        ;If not, return carry set
209F: A9 CC      249  DOTYPE4  LDA  #$CC         ;MLI close code
20A1: 4C 70 BE   250           JMP  GOSYSTEM     ;Close file, return proper carry
20A4: A0 00      251  DOTYPE5  LDY  #0           ;Initialize index
20A6: B9 00 02   252  DOTYPE6  LDA  INBUF,Y      ;Fetch character
20A9: 09 80      253           ORA  #$80         ;Insure high bit set
20AB: 20 ED FD   254           JSR  COUT         ;Display character
20AE: C8         255           INY               ;Advance index
20AF: CE DB BE   256           DEC  RWTRANS      ;Decrement input count
20B2: D0 F2      257           BNE  DOTYPE6      ;If not done with this record
20B4: AD 00 C0   258           LDA  KBD          ;Key pressed?
20B7: 10 DB      259           BPL  DOTYPE3      ;If not, get next record
20B9: 8D 10 C0   260  DOTYPE7  STA  KBDSTRB      ;Kill key strobe
20BC: C9 83      261           CMP  #$83         ;[Ctrl-C] to abort?
20BE: F0 DF      262           BEQ  DOTYPE4      ;If yes, abort
20C0: C9 93      263           CMP  #$93         ;[Ctrl-S] to pause?
20C2: D0 D0      264           BNE  DOTYPE3      ;If not, get next record
20C4: AD 00 C0   265  DOTYPE8  LDA  KBD          ;Key pressed?
20C7: 10 FB      266           BPL  DOTYPE8      ;If not, wait
20C9: 30 EE      267           BMI  DOTYPE7      ;Yes, process it

                 269  TYPEND   =    *

--End assembly, 362 bytes, Errors: 0
```

wise the system may become a bit confused.

The rest of the TYPE code is an exercise in using the MLI. In simple terms, this code opens the file, reads records, displays records, and finally closes the file. The well commented code should assist the reader in understanding the logic of this process.

The code makes use of standard MLI buffers located in the BASIC.SYSTEM Global Page. The easiest way to do this is use the GOSYSTEM subroutine located in the BASIC.SYSTEM global page. This subroutine is entered with the MLI code in the A register. If no error occurs, carry is returned clear. If an error does occur, carry is returned set and the A register contains a number between 2 and 21, a standard error number.

If one chooses this route, they must gain a good understanding of how BASIC.SYSTEM uses the buffers. A more straight forward and shorter version of TYPE used MLI buffers within the code itself. However, this made the position in-

dependent nature of the code somewhat more difficult. Also, the TYPE code was over 200 bytes longer.

There are some observations that must be made regarding the MLI. The first of these involves the PREFIX. When BASIC.SYSTEM is booted into memory and begins operation, the PREFIX will be a null string. If special processing is not performed, a null PREFIX will cause an Invalid Pathname Syntax Error message to be returned for something like an OPEN through the MLI.

There are several ways around this problem. The STARTUP program could create a non null PREFIX via the statement

```
PRINT CHR$(4);"PREFIX":INPUT
PR$:PRINT CHR$(4);"PREFIX
";PR$
```

which sets the PREFIX to the volume name of the disk in Drive 1. This can be performed under MLI control quite easily. If PFIXPTR at $BF9A is not zero, a PREFIX is active, therefore, there is no problem.

If PFIXPTR is zero, a null PREFIX exists. Using the last slot and drive information in DEVNUM ($BF30), an ON LINE call to the MLI returns the volume name based on the slot and drive information. This volume name is then used in a SET PREFIX call to the MLI.

Under the user command environment, setting Bit 3 in PBITS+1 tells the system to produce a pathname in $280 that begins with the volume name of the currently active drive. On the other hand, if the user explicitly enters a prefix in the path name, the volume name is not inserted into the pathname. Simply setting Bit 3 saves some 125 bytes of code required for TYPE. This proves the point that serious users of user command code must gain intimate knowledge of what PBITS and FBITS provide.

Another observation involves buffer management. When a file is opened, a 1K buffer is required. Since this is a temporary buffer for the life of the OPENed file, it would seem quite logical to dynamically obtain this buffer via GETBUFR. After the file is closed, in theory, FREEBUFR ($BEF8) deallocates all buffers obtained via GETBUFR. This works with an unexpected result.

FREEBUFR returns HIMEM set at $9600 which effectively mangles all or part of user command code.

Subsequent study of the code for BASIC.SYSTEM has provided a solution to this problem. Based on the 1-Jan-84 release of ProDOS, $BEFB contains the page number that FREEBUFR used to produce the final HIMEM value. The default is $96. Using TYPE as an example, after GETBUFR returns the page number of the beginning of the requested "buffer," subtract 4 from that number and store it in $BEFB. Now use GETBUFR and FREEBUFR as often as desired with the end result of the TYPE code never being disturbed. In this example, the new default for HIMEM is $95, if no other user commands are installed.

Installation of TYPE

First consider the user that enters the source code for an assembler such as BIG MAC LC. Assemble the program, convert it to ProDOS using CONVERT (if the ProDOS assember is not used), then BRUN TYPE to install the command. Or in the STARTUP program add the statements

```
PRINT CHR$(4);"BLOAD TYPE"
CALL 8033
```

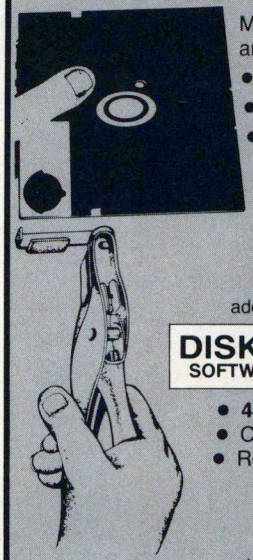
Now a crucial step. Immediately after installing TYPE as described above, try the LIST command in immediate mode. If the disk fires up and ?SYNTAX ERROR occurs, this is due to a bug in ProDOS. The system must be re-booted and before any commands are installed, execute only once the statement:

```
POKE 48647,PEEK(48647)-1
```

The logical place for this statement is the very first statement of the STARTUP program. For the hackers, before any command is installed, $BE06 jumps to an RTS instruction at $BE9E. It should jump to the SEC instrucion at $BE9D.

For the user entering only the hex code, the process is a bit more cumbersome. Under monitor control, enter the code as listed starting at $1F61. Now perform a BSAVE TYPE,A$1F61,L362. The CONVERT step is not necessary since it is assumed that you are using ProDOS. Now you can BRUN TYPE or use the BASIC statements given above to install TYPE.

*REDUCES DISKETTE COST 50%!



Make use of the back of your 5¼" Diskettes and SAVE

- NIBBLE NOTCH tools make it easy.
- Adds the precise notch where you need it.
- Doubles diskette space or MONEY BACK!

NIBBLE NOTCH I
Cuts square notch for
Apple, II, II+, IIe, III,
Franklin and Commodore.
only $14.95 each
add $1.50 each P&H ($4.50 each foreign P&H)

DISK OPTIMIZER© SYSTEM
SOFTWARE FOR APPLE, II, II+, IIe, III & Franklin

- 469% FASTER Than Similar Programs!
- Certifies your "new" disk 100% Error Free
- Removes Bad Sectors • Adds 36th Track
- Performs Disk Drive Speed Check
- Adds DOS and More

only $24.95 each
add $1.50 each P&H ($4.50 each foreign P&H)

— OR BUY BOTH —
only $29.95
add $2.50 each set P&H ($6.50 foreign P&H)
Florida Residents Add 5% Sales Tax

SATISFACTION GUARANTEED OR YOUR MONEY BACK!
ORDER TODAY!
SEND CHECK OR MONEY ORDER TO:
NIBBLE NOTCH COMPUTER PRODUCTS
DIVISION OF CORTRAN INTERNATIONAL
4211 NW 75th TERRACE • DEPT. 7 6 • LAUDERHILL, FL 33319
PAT. PEND. ALL TRADEMARKS ARE ACKNOWLEDGED

Circle 4 on Reader Service Card

68000 for the Apple II+ //e
Gnome Card

A new low-cost 68000 co-processing board is available now!! The Gnome 68000 co-processing board is a new development tool. Now users can move into the new technology for a low price. The Gnome board has an expansion interface connector for expandability. Your current Apple can develop software for the 68000 using the Gnome board. You can learn 68000 assembly language and interfacing with the expansion connector on the Gnome board. Fast math packages can be put in rom and called by Applesoft or assembly. The Gnome board can do faster graphics than the Apple 6502 can ever do!!

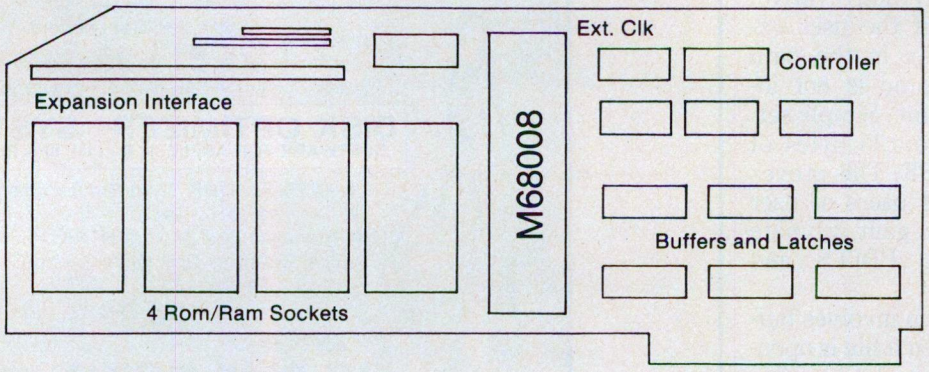
Features:

- **True Co-Processing M68008 at 7mh. Apple and Gnome at *full-speed.***
- **Gnome card has up to 64k rom/ram at 7mh, *no* wait states.**
- **Gnome uses the Apple 64k ram at 1Mhz unlike other cards!**
- **Comes with 8k Monitor rom, 2k static ram. Stepper debugger in rom.**
- **Gnome can be turned on/off from Applesoft or 6502 assembly.**
- **Apple can interupt Gnome to prompt for action/performance.**
- **Use Apple Dos 3.3 *from the Gnome Monitor* to load/save files!**
- **Can use any extended memory boards made for the Apple Computer.**
- **Develop programs on Gnome and send over to the Macintosh!!**

We have the capability *now*, before Apple's Mac Assembler!

- **Exception Handing Capability**
- **Stand Alone Capability**
- **Slot Independent!**
- **Demo Programs and Source Listing**
- **Full Macro Assembler Available**
- **Expansion Bus Interface**
- **Can Expand To 1Mbyte Memory**

Many more .....



Call or Write:

MA *Micro-Analyst, Inc.*
P.O. Box 15003
Austin, Texas 78761
(512) 926-4527

Dealer Inquiries Welcome!

Regular Price $295.00
Special Introductory Price $240.00

Please call to order at ***limited* Special Price!**

Apple is a Registered Trademark of Apple Computer.

Circle 16 on Reader Service Card

ProDOS — Pros and Cons

Cornelis Bongers

PRODOS

IN the past few months I had the opportunity to work extensively with the new ProDOS from Apple Inc. Although I am certainly not yet familiar with all the ins and outs of ProDOS, the things I have seen so far are very impressive, to say the least. One of the most attractive points is that loading of all kinds of files is speeded up considerably. Furthermore, there is now a fast garbage collection command and a real CHAIN command for Applesoft programs.

Because ProDOS traps every executed Applesoft statement, garbage collection will be done as soon as the amount of free memory falls below a certain limit. This means that in principle you do not need to specify PRINT CHR$(4);"FRE" statements in your program, because ProDOS takes care of garbage collection and prevents Applesoft from executing its (slow) garbage collection routine. The availability of a good working CHAIN command (in combination with the quicker loading of programs) will have a major impact upon the size of software packages that can be implemented on the Apple. It is now possible to split up programs in two or more parts and to load and execute each in turn.

Another improvement concerns the hierarchical directory structure. This is of most interest to hard disk users with 80 track drives, but those who use the standard Apple drives will also find this feature handy. As to the "under the hood" aspects of ProDOS, the availability of a well documented set of MLI calls, combined with the clearly defined "system global page" and "basic global page" is very valuable for software developers.

The global pages contain relevant system information, among others a list of disk device driver vectors, a list of buffer addresses, and interrupt routines and information about the system attributes. This eases, for example, the task of connecting a non-Apple device to ProDOS. Since all communication between ProDOS and the devices is done via the device driver vectors, connecting an 80 track drive no longer requires DOS patches. Furthermore, it took only one day and no ProDOS patches to implement a RAM disk for the BASIS 108, despite the fact that the BASIS employs an entirely different bank-switching system.

When compared with DOS 3.3, the directory structure of ProDOS allows a lot of freedom. As far as I could ascertain, only the volume directory must always be stored in block 2. The remaining directory blocks (or sub-directory blocks), as well as the volume bit map may be stored in user-selected blocks. Of course there are also some disadvantages related to the use of ProDOS. ProDOS consumes 16K more (language card) memory in comparison with DOS 3.3 (including the memory areas reserved for future expansion). This means that users of DOS MOVER will effectively loose some 10-12K for their application programs.

Another disadvantage of ProDOS is that sometimes more work than necessary need be done to execute simple operations. For example, when you copy a file with the FILER, you have to do a lot more typing (i.e. entering volume names) than with FID. I would not consider this a real disadvantage though, since it will probably not take long before someone writes a copy program that requires less typing effort from the user.

A final point worth mentioning is that I had some problems with the block read routine. Since similar problems were reported by other users, there may be some kind of bug there. What happened is that after working a while with ProDOS, one of the (Apple) drives started to behave strangely. Upon a CAT command the drive made a clattering noise, which only stopped after I opened the door of the drive. Neither rebooting, not turning the power off/on helped. The only way to get the drive in line again was booting a DOS 3.3 master, executing a CATALOG command and rebooting ProDOS. After that, the drive behaved normally again. I suspect that under certain circumstances the sequence of head positioning instructions given by the block read routine causes a "resonance" in the drive.

Taking the few bugs in ProDOS for granted, my overall conclusion is that ProDOS is really an excellent operating system and I am sure it will not take long before most Apple users will work in the ProDOS environment. At the University, most of us do already.

The BASIS 108 versus the Apple *//e*

As you will have deduced from the title, ProDOS is not the only matter of interest here. I will use the remainder of this note (if the reviewers approve) to say a few words about the BASIS and the *//e* and discuss how to implement ProDOS on the BASIS 108. First about the BASIS and the *//e*. By the way, when I am talking about the BASIS 108, I mean a BASIS with Applesoft and the 80 column monitor in ROM.

Yesterday I gave a lecture on Pro-

DOS and one of the questions (of course) was: "How can we implement ProDOS on the BASIS 108?" Since, I have a BASIS myself, this question was readily answered (see below). The next question however was harder to answer : "What are the advantages of a BASIS 108 relative to the Apple *IIe*?" This put me in a somewhat difficult position, for I am an enthusiastic user of the Apple][+ and all kind of Apple products, but so far I managed to keep my opinion about the *IIe* in the private domain. This despite all temptations, such as the article from my fellow staff writer Rick Sutcliffe about Genetic Engineering (*Call -A.P.P.L.E.* Vol VI, Number 12, Dec 1983). Anyway, maybe it is worth trying to get a different opinion about the *IIe* published.

In the Genetic Engineering article, the BASIS 108 is compared with the *IIe* and to my great surprise the *IIe* got a slightly higher overall rating. Personally I do not agree with this. In my opinion, the *IIe* ought to have a considerably *lower* rating than the BASIS 108.

Let's examine a few aspects more closely. First innovativeness, for which the BASIS gets 8.5 points and the *IIe* 9 points. Well folks, from the user's point of view, there is nothing on the *IIe* that is really innovative. The 80 column screen for example is implemented in the worst thinkable way. You still have to fiddle around with PR#3 commands and POKEing 36 or 1403 seems to be the only way to get your TAB's correct. To make matters worse, you are sure to have problems when you enable a printer and the 80 column card is active.

Another nice example of how things should not be done is the inverse blank line you get when the screen scrolls and the inverse flag is set. I do not need to go into further detail, for the size of the 80 column manual for the *IIe* speaks for itself. Anyway, I stopped taking things seriously after encountering the sentence that I feel is characteristic for the 80 column card design:

> "The 80-column card is not designed to support the standard Applesoft INPUT and GET statement, nor the underlying monitor KEYIN routine. Use of them can result in loss of program or data; use the blinking underline cursor routine or read HEX $C000 directly" (yes, this is really in the manual).

On the BASIS 108, 80 column mode is implemented as it should be: no PR#3, no POKE 36's, no PEEKings of $C000, no problems. The only *mistake* made by the designers of the BASIS is that they overlooked that programmers often access the screen directly (with LDA or STA ($28),Y). This technique (a better word is "trick") fails on the BASIS (but also on Apples with an 80 column screen). A good solution has now been found by a dealer in the Netherlands who equips the BASIS with a switchable monitor EPROM. When you have programs that access the screen directly, you switch on the 40 column monitor, otherwise you can use the 80 column monitor.

Another point of innovation concerns the second 64K bank. On the BASIS 108, this bank is entirely free for the programmer. Those who develop software for the *IIe* on the other hand, encounter a part of the 80 column screen in that bank. Really, I couldn't believe my eyes when I saw the *IIe* memory map in the documentation. One of the unavoidable consequences of this setup is that on the *IIe* you have only 120 free blocks on your RAMdisk (/RAM), whereas on the BASIS 108 you have 126 free blocks.

If the designers of the *IIe* really had been innovative, they would at least have copied the BASIS design, where a separate 2K memory bank is reserved for the 80 column screen and the RAMdisk driver. The only thing that is done better on the *IIe* is the implementation of the read/write options (you can write to one bank while reading from another) and of course the availability of the double high-res screen.

Having dealt with innovation, we proceed with faithfulness to the][+. The BASIS gets an 8 and the *IIe* a 9.8. In my opinion, the BASIS is more compatible with the][+ than

Highest Quality, **Lifetime Guarantee!**

DISKETTES

$1.65 5¼" soft-sectored, hub ring, envelopes, double density, double-sided on APPLE drives -- 100 for $155, 100 single-sided for $149.

Hard plastic stand-up 10-diskette carrying cases $2.75 each, 4 for $10 (beige, black, blue, green, grey, red, yellow). Smoked-plastic flip-top 75 diskette file cases, $19.50. Heavy-duty nibbling tool, $22.

Disk Drives

$199 100% APPLE-compatible, 40-track, full-size, Siemens type quality drives, with manufacturer's 1-year warrantee. Controller card, $65.

COD & VISA/Master Card orders welcome. Add $4 for shipping & handling (only $2 for orders under $50) plus 6% sales tax for DC residents. Send for our catalog.

VF ASSOCIATES

Western Ave., N.W., Wash., D.C. 20015 (202) 363-1313

Circle 36 on Reader Service Card

the *IIe*. One of the first programs I tried on the *IIe* was LOCKSMITH 4.1 and, although I didn't get a system crash, the drive speed option I selected refused to display any information except for a picture of a padlock. The BASIS/Apple][+ compatibility, on the other hand, is 100% for software that doesn't use the language card. The reason is that you can load Applesoft plus the Apple monitor in the RAMcard and next you can disable all language card switches. The software then cannot switch off the language card anymore, and since everything looks and works as a][+, the software will run flawlessly. Most software that needs a language card will also work fine on the BASIS, but it is true that sometimes adjustments are needed. If you use the switchable monitor mentioned above, you are assured of a very high degree of compatibility again. Those who applied my Applesoft patches (to let Applesoft "support" a 40 as well as an 80 column monitor) have nearly 100% compatibility, the exceptions being programs that check on the presence of the original Applesoft. So far I have not encountered such programs.

As to the physical considerations and the keyboard, I generally agree with Rick, but I think that the cursor block on the BASIS should have been mentioned explicitly. It goes without saying that this cursor block works much more conveniently than the cursor keys on the *IIe*. Furthermore, in the BASIS, pressing escape to initiate cursor movement is not necessary. In addition, apart from the cursor movement keys, there are also keys to clear the screen, to clear to the end of screen, and to clear to the end of line. Of course, the *normal* Apple][+ arrow keys are included in the cursor block.

The special feature section is not rated separately (the BASIS has standard 80 column, 128K, serial and parallel port and Z80), but I would weight the extra's heavily, since the (list) price difference between the BASIS 108 and the *IIe* with the extended 80 column card is (in the Netherlands) only $150. For that money you get two interfaces and the Z80 with CP/M, but you must buy the Applesoft ROMs. It can therefore come as no surprise that given the option to buy a BASIS 108 or an Apple *IIe* (in its current state) I would not hesitate to choose the BASIS again. My grand total rating is 9 points for the BASIS 108 and 7.5 points for the *IIe*.

Now, before the war between the U.S.A. and the Netherlands starts, I want to state clearly that the choice of a computer should not be determined by hardware considerations alone. Therefore I can have peace with Rick's conclusion: "I have to give the overall edge to the *IIe* because it has already an established place as the market leader" (in view of the success of the IBM PC, the term market leader ought perhaps be double quoted). This is quite true and relevant for most users, since software — as well as hardware support for the *IIe* is much better than that for the BASIS. However, for me this does not mean that as a consequence the *IIe* is also the better machine, it is only better regarding this aspect. Everyone who has a reasonable knowledge of the Apple][+, Apple software, and machine language will enjoy the BASIS much more than the *IIe*. On the other hand, those who mainly use their machine to run all kinds of software packages probably should buy a *IIe* (or an IBM PC), for in this respect they obtain maximum support from the software industry as well as from the magazines.

ProDOS on the BASIS 108

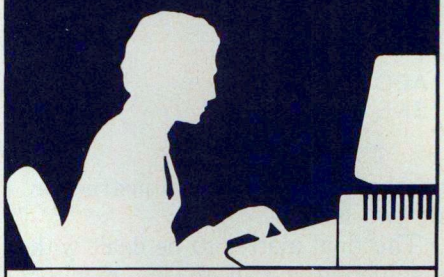
The final point to be discussed in this note concerns the ProDOS implementation on the BASIS. Those who have a BASIS (with Applesoft and the BASIS 80 column monitor in ROM) and bought ProDOS will by now have discovered that ProDOS doesn't boot on the BASIS 108. I attacked this problem with William Schouten, and we discovered that the reason is twofold. In the first place, there is a routine in ProDOS that determines the type of the Apple (i.e.][,][+ or *IIe*) and this routine cannot classify the BASIS into one of the existing Apple categories. The easiest way to deal with this problem is to search the ProDOS disk with a zap utility for the instruction:

```
LDX $FBB3 (AE B3 FB).
```

Change this instruction into:

```
LDX -$EA (A2 EA)
NOP      (EA)
```


EDX/AB...
if you are serious about programming



EDX/AB is a full screen Applesoft editor which turns the Apple II+ or *IIe* into a serious programming tool. Full screen editing allows you to make changes to the program anywhere on the screen, unlike line editors which limit you to calling up and changing only one line at a time. In addition you can scroll through the program in both directions—forward and backward either a screenful at a time or line by line. Gone is that helpless feeling of watching the line you want roll off the screen. A simple keystroke brings it back.

EDX/AB is designed to work with ProDOS, Apple's new disk operating system, and has such "wouldn't it be nice" features as:

- full search and replace capabilities, either global or item by item.
- individual and total cross referencing for variables and GOTO, GOSUB.
- complete control of move, copy and deletion of lines (the move is a 'smart' move where all references to the moved lines are automatically adjusted).
- an active variables list which displays current values, including arrays.
- an extra fast renumber, partial or complete, that works up to five times faster than Apple's renumber.

EDX/AB does this and more. Hex to Dec, Dec to Hex is there of course, as are many additional helpful commands. EDX/AB even provides selectable auto line numbering. In all there are 25 commands, easily listed on the 'Help' screen. The commands are so simple that, after reading the manual, the 'Help' screen will be the only reference you'll need. Just two keystrokes implement any command, and they are at your fingertips all the time, so there is no repetitive loading in from disk.

At TROY SOFTWARE we felt it was time someone took the frustration out of Applesoft programming.

EDX/AB™ $69.95

Contact your dealer or order direct.
(Visa and Master Charge orders accepted)
Please add $2.50 for shipping and handling.
Please specify APPLE II or APPLE *IIe*.

TROY SOFTWARE DESIGN INC.
63 Roseview Avenue, Richmond Hill, Ontario
Canada L4C 1C6 ☎(416) 884-4527

* ProDOS requires 64 K RAM and APPLESOFT in ROM.
APPLE and ProDOS are registered trademarks of APPLE Computers Inc.

Circle 33 on Reader Service Card

and your BASIS will be classified as a][+. The ProDOS master still will not boot, because a check is also done on other monitor locations. Fortunately, this check can also be neutralized easily. Just search for the instructions:

```
ADC #$0B (69 B)
BNE +5   (D0 3)
```

and change the BNE +5 instruction into two NOP's.

The final matter to be dealt with is adjusting a call to the monitor READKEY routine. BASIC.SYSTEM uses JSR $FD10 (quite strange since this is a non-documented monitor entry), but on the BASIS, you should use JSR $FD18 (20 18 FD). So, change all JSR $FD10's (20 10 FD) to JSR $FD18's (occurs twice in my latest version of ProDOS). Finally, search the disk for all occurrences of the instruction: STA $C00F (8D 0F C0) and change each of these into three NOP's. Now when you boot the ProDOS disk, things should work nicely. There certainly is room for improvement here and there, but I worked for three months with these patches without encountering any serious problems. Two small improvements you may want to make:

a) Up (down?) grade your BASIS to a *IIe*. This can easily be done by setting the MACHID byte to $A0. I did not test this modification extensively, so take care. It is strongly advised to leave the 80 column card bit clear, otherwise nasty things may happen.

b) Adjust the FILER so that it is able to read ("pick") screen characters when you use the forward arrow key. Yes, the FILER does indeed what is forbidden by all programming rules, for it stubbornly reads characters directly from the screen (with LDA ($28),Y) and this of course leads to problems on the BASIS. You can change this by first BLOADing the FILER in memory and next searching for the instruction sequence:

```
LDY $24 (A4 24)
LDA ($28),Y (B1 28)
```

Change this sequence to:

```
JSR $FEFE
NOP
```

Finally, BSAVE the FILER (the start and length of BLOADed files can be retrieved from locations $BED7-$BEDC, see the ProDOS Technical Reference manual). By the way, I looked on the ProDOS disk and noticed that the LDA ($28),Y instruction occurs rather frequently. Probably, the operation of other programs (in particular EDASM) can also be improved by making adjustments similar to the one above.

The $175 disk drive for Apple II

Now, you can have a full-sized 5¼" floppy disk drive compatible with the Apple II for the unbelievably low price of $175.
Also, check out these half-height 5¼" floppies:

- Single Apple half-height $195.00
- Dual Apple half-height $390.00

(includes the drive, cabinet, cable, and our standard warranty including 90 days parts and labor)

AVAILABLE NOW FOR DELIVERY

TELEPHONE ORDERS ONLY
MASTERCARD, VISA, CASHIER'S CHECKS COD ORDERS ACCEPTED
(818) 883-0587

StarLogic

20860 Plummer Street
Chatsworth, CA 91311

Apple is a registered trademark of Apple Computer, Inc.

(Also compatible with Franklin ACE)

Circle 12 on Reader Service Card

THE SPREADSHEET SHEET:

Data Analysis

(PART 6)

Jeffrey M. Jacques, Ph.D.

Introduction

WELCOME back. Last time we completed our discussion of statistical procedures which may be used to summarize the joint distribution of data when such data are either (1) categorized with the use of a crosstabulation procedure or (2) summarized by the linear regression equation. We noted the importance of examining the proportionate reduction in error — the degree to which we can reduce the errors in predicting the variable of interest when we know the joint distribution of the variable of interest and the explanatory variable(s). All of these statistics were developed from appropriate formulae. But there is an additional logical question: Given what I know now about my random sample, how much faith can I have in these statistics when I wish to say something about the population?

All along we have noted that this set of twenty faculty cases was a random sample from a larger college population of several hundred faculty. We are now interested in the larger question of evaluating our faith in these summary statistics, and the relationships we have found among the variables, for the population. The specific question becomes: Are the conclusions we have drawn about the variables based on our sample of faculty likely to be true when we consider all the faculty from this college? To answer this question, we must explore the ideas behind inferential statistics.

The Fundamentals of Inferential Statistics

Effective use of inferential statistics requires the use of: (1) the principles of probability, (2) our knowledge of the behavior of certain statistics, and (3) the logic of statistical tests of significance. Whenever we are working with a random sample, we must expect that the data are similar to, but not exactly the same as, the population parameters under investigation. This difference, the difference between the sample statistic and its population parameter, is typically called sampling error. Most of the time, when a true random sample is selected from a known population, there is generally only a small difference between the sample statistics and the population parameters. When there are substantial differences, these differences may not be due to sampling error, but may be due to bias in the sample selection process. An example will help.

Let's assume that you and four of your friends went to this college and each selected a sample of 20 faculty members. All of your friends claimed that they selected the sample in a random manner. Each of you then determined the sex of each faculty member and develop a simple frequency distribution. You pool your information and find that each of you had a different number of male and female faculty. Table 6.1 summarizes these results.

The number of female faculty ranged from a low of 2 to a high of 10. If we know from other sources that the true population distribution was 40% female and 60% male, then in a random sample of 20 faculty, we would expect approximately 8 (i.e., .4 * 20 = 8) female faculty and 12 male faculty. But only one person had this exact number, one person had more than the expected number of female faculty, and three people had fewer than 8 female faculty. You are then left with the question: At what point do I attribute the differences between my sample statistic and the popula-

TABLE 6.1

HYPOTHETICAL DATA FROM 5 INDEPENDENTLY SELECTED SAMPLES

Faculty Sex Code	First Sample	Second Sample	Third Sample	Fourth Sample	Fifth Sample
Female	10	8	6	3	2
Male	10	12	14	17	18
Total	20	20	20	20	20

tion parameter as minimal sampling error, or, at what point do I conclude that there was bias in the sample selection process?

One of the better and more important approaches to resolving this question is through the ideas of probability and inferential statistics. Two questions or hypotheses may be stated in order to determine if the sample statistic is reliably different from the overall population.

Ho: There is no significant difference between the sex distribution of the sample and the sex distribution of the population.

H1: There is a significant difference between the sex distribution of the sample and the sex distribution of the population.

The first hypothesis is typically called the null hypothesis. This is pretty reasonable since it contains the idea of no difference. The second hypothesis is typically called the alternate hypothesis. The alternate hypothesis specifically contradicts the null hypothesis.

Well, which hypothesis is true? There is no direct answer. But, we can answer the question of which hypothesis is most likely to be false on the basis of probability. Although there is nothing sacred about choosing any probability level, scientists typically use the .05 or .01 probability levels. Working with the .05 probability level, we would not wish to incorrectly reject the null hypothesis more than five times in 100 when in fact there are no differences. This is called Type I error — rejecting the null hypothesis when it is actually true, and the probability level is typically referred to as the level of statistical significance. This may sound a little confusing at first, but you will get used to it once you see its utility.

CHI Square and Tests of Statistical Significance

When we work with true random samples, we know quite a bit about how frequently one would find differences between sample values (e.g., observed patterns) and population parameters (e.g., expected patterns) of any data set. Indeed, a specific distribution known as the chi square distribution can be used to compare how likely varying differences between observed and expected patterns in the data occur simply by chance. All you need know are: (1) the computed chi square value, (2) the likelihood or probability of finding a given chi square value, and (3) the number of degrees of freedom. The number of degrees of freedom is equal to the number of independent equations – 1. Once you know the total number of cases, then any single missing cell value may be found with only one equation. We can quickly determine the number of degrees of freedom by counting the total number of different categories or attributes. Since sex has two attributes:

$$\text{degrees of freedom} = 2 - 1 = 1$$

We noted earlier that we would use .05 as the level of significance; that is, we are not interested in rejecting a true null hypothesis more often than five times in 100. We can use these two pieces of information and apply them to the chi square distribution to determine the critical value of chi square from its known distribution. Any basic statistics textbook will have a chi square table which includes critical values. In this case, the critical value (i.e., df = 1, significance level = .05) would be 3.841.

We may now use the chi square formula of:

$$\text{Chi Square} = \sum_{i=1}^{N} \frac{(\text{observed} - \text{expected})^2}{\text{expected}}$$

For each attribute, compute the difference between the observed and expected cell frequencies, square this difference, then divide by the expected cell frequency. Once the contribution of each cell is determined, simply sum these values.

Since, as we noted earlier, the population distribution was 40% female, we would have expected 8 female and 12 male faculty. Using the data from Table 6.1, the first random sample, we can substitute the values in the above formula to yield:

$$\text{chi square} = \frac{(10-8)^2}{8} + \frac{(10-12)^2}{12}$$

$$= .5 + .3333$$

$$= .83333$$

Since the computed value of chi square (i.e.,.8333) was less than the critical value (i.e., 3.841), we cannot reject the null hypothesis. The first, second (which had 8 female faculty), and third (which had 6 female faculty — 6 from 8 will yield the same chi square value as 10 from 8) random samples did not have very different results from that expected simply by sampling error. For sample number 4, however:

$$\text{chi square} = \frac{(3-8)^2}{8} + \frac{(17-12)^2}{12}$$

$$= 3.125 + 2.08$$

$$= 5.205$$

This large chi square value indicates some type of bias rather than simple sampling error. And if this distribution leads us to conclude some type of bias, every more skewed distribution (e.g., number of female faculty = 2) must have fallen outside the range of simple sampling error (e.g., chi square (1 df) = 7.5; sig. < .01). We could determine if our sampling technique was biased by examining the distribution of other variables (e.g., rank, degree, etc.). All we need to know are the parameters of these variables for this college.

Constructing a Template for Testing Sampling Bias

In the past several articles we have spent most of our time modifying worksheets. This time we will create a separate template which may be used to test for sampling bias. In order to complete this template so that it is efficient, we shall use the following functions:

1. @IF
2. @NA
3. @OR
4. @ISNA
5. @SUM
6. @MAX
7. @LOOKUP
8. @ERROR

Since most of the variables you will be dealing with have five or fewer attributes or categories, then a master data entry table may be constructed. Additionally, we will identify specific areas of the worksheet for computational space and tabular values. Given the algebraic complexity of finding the significance value for a computed value of chi square, we shall construct a table of chi square critical and significance values. Therefore, the top part of the tem-

plate shall be reserved for data entry, the middle for computational equations, and the bottom for the lookup table.

The data entry table was constructed so that it may handle a maximum of five attributes or categories. For each attribute of the variable under consideration, you must enter the observed and expected values. The data from this table — the sum of observed and expected frequencies — are then checked, and essential statistics are automatically recalculated with two presses of the ! key. In order to do this efficiently, each cell of the data entry portion of the template has the function @NA. This function is useful since it is treated as a value and, therefore, may be checked with an appropriate @IF statement. That is, in the computational portion of our template, the following @IF statement (for cell C20) may be used:

@IF(@OR(@ISNA(C8),
@ISNA(C10)),0,C8)

Cell C8 contains the observed frequency from the first attribute and cell C10 contains the expected frequency. If either (@OR) of these cells were not replaced with a numeric value and still contained @NA, then *THE Spreadsheet* will return a value of zero. If there is a numeric value in cell C8 and C10, then cell C20 will have the same value as that of cell C8. Similarly, all the observed and expected frequency cells are checked with appropriately modified @IF statements and the appropriate values are placed in cells C20 through G22. Then, in cell H20 and H22, the number of observed values and expected values are summed, respectively. To reduce the likelihood of an error, the following equation (in cell H13) checks to make sure that these summations are equal:

@IF(H20<>H22),@ERROR,@NA)

If they are unequal, then ERROR will

Table 6.2

	A	B	C	D	E	F	G	H
1			Chi Square Test of Sample Representativeness					
2	==========							
3								
4			Data Entry Table					
5	----------							
6			Attr. #1	Attr. #2	Attr. #3	Attr.#4	Attr. #5	
7								
8	Observed frequency		@NA	@NA	@NA	@NA	@NA	
9								
10	Expected frequency		@NA	@NA	@NA	@NA	@NA	
11	----------							
12								
13	Chi^2 with (	+H29	df) =	+H24	; sig. =	@MAX(C31...H31)	@IF(H20<>H22,@ERROR,@NA)	
14								
15	==========							
16	A	B	C			F	G	H
17			Computational Space					TOTAL
18	----------							
19								
20	Observed frequency		@IF(@OR(@ISNA(C8),@ISNA(C10)),0,C8)		...	@IF(@OR(@ISNA(G8),@ISNA(G10)),0,G8)		@SUM(C20.G20)
21								
22	Expected frequency		@IF(@OR(@ISNA(C8),@ISNA(C10)),0,C10)		...	@IF(@OR(@ISNA(G8),@ISNA(G10)),0,G10)		@SUM(C22.G22)
23								
24	Cell contribut'n to X^2		@IF(C22=0,0,(C20-C22)^2/C22)		...	@IF(G22=0,0,(G20-G22)^2/G22)		@SUM(C24.G24)
25								
26								
27	N. of valid cells		@IF(@OR(@ISNA(C8),@ISNA(C10)),0,1)		...	@IF(@OR(@ISNA(G8),@ISNA(G10)),0,1)		@SUM(C27.G27)
28								
29	Degrees of freedom						=)	+H27-1
30								
31	Significance Level		@IF(H29=1,@LOOKUP(H24,A40...A52),0)		...	@IF(H29=4,@LOOKUP(H24,G40...G52),0)	=)	@MAX(C31.F31)
32								
33	----------							
34	A	B	C	D	E	F	G	H
35		Look-up	Table: Chi Square Statistical Sig. Level					
36	----------							
37	Where df = 1		Where df = 2		Where df = 3		Where df = 4	
38	Comp X^2	Sig.	Comp X^2	Sig.	Comp X^2	Sig.	Comp X^2	Sig.
39	----------							
40	0	1	0	1	0	1	0	1
41	.000628	.98	.0404	.98	.185	.98	.429	.98
42	.00393	.95	.103	.95	.352	.95	.711	.95
43	.0158	.9	.211	.9	.584	.9	1.064	.9
44	.0642	.8	.446	.8	1.005	.8	1.649	.8
45	.148	.7	.713	.7	1.424	.7	2.195	.7
46	.455	.5	1.386	.5	2.366	.5	3.357	.5
47	1.074	.3	2.408	.3	3.665	.3	4.878	.3
48	1.642	.2	3.219	.2	4.642	.2	5.989	.2
49	2.706	.1	4.605	.1	6.251	.1	7.779	.1
50	3.841	.05	5.991	.05	7.815	.05	9.488	.05
51	5.412	.02	7.824	.02	9.837	.02	11.668	.02
52	6.635	.01	9.21	.01	11.341	.01	13.277	.01
53	----------							
54	NOTE: Press ! twice. Set column global column width to 9.							

Columns do not align. Remember to /Reproduce the missing statements for columns D and E.

appear in cell H13. If they are equal, NA will continue to appear. See Table 6.2.

Since we know: (1) the formula for computing the contribution of each cell to the overall chi square value, and (2) when a zero exists in the expected cell frequency row the cell contribution cannot be computed, then we can determine for each attribute their contribution to the overall chi square value with the following equation:

@IF(C22 = 0,0,(C20-C22)^2/C22)

If cell C22 contains a zero, then return a value of zero to cell C24. If this cell contains a value other than zero, then return its contribution to the overall chi square value. (By /Reproducing this @IF statement from cell C24 through cell G24, answering Relative to the computer queries, you may save considerable time and energy.) Then @SUM the values across this row to find the overall chi square computed value. Simply place @SUM (C24...G24) in cell H24. Now, that's not too tough.

In order to determine the appropriate degrees of freedom, we must first compute the number of valid (e.g., in this case, those cells without @NA in them) cell pairs (observed and expected frequencies). The @IF statement (for cell C27) should read:

@IF(@OR(@ISNA(C8),@ISNA (C10)),0,1)

If either cell C8 or C10 has the @NA function, then return a zero, otherwise return a 1. By @SUMming the number of 1s, we can count the number of valid attributes. The number of degrees of freedom may then be computed from this summed value. See rows 27 and 29 of Table 6.2 for more information.

The final computational step — determining the level of significance — may be accomplished most easily with the @IF and @LOOKUP functions. Given two columns of numbers (e.g., the first an ordered list of critical chi square values and the second its corresponding level of significance), the @LOOKUP function returns the value in the second column (e.g., level of significance) which is the maximum value equal to, but not greater than, that value in the first column (e.g.,critical chi square value). Since we may have as many as four degrees of freedom, four sets of two columns must be created. Using chi square tabular values (see Blalock, 1979), we may compare our computed chi square value to these critical values and return the level of significance with the use of the following @IF statement:

@IF(H29 = 1,@LOOKUP (H24,A40...A52),0)

Cell H29 contains the number of degrees of freedom. If it is 1, then use column A, rows 40 through 52 to lookup and compare the computed chi square value stored in cell H24 and return the appropriate value from column B. (e.g., with H24 = .8333, and H29 = 1, then .5 (cell B46) would be returned to cell C31.) By /Reproducing and /Editing, we can check to see if the degrees of freedom are 1, 2, 3, or 4. Then, using the correct pair of columns, we can determine the level of significance. Once this value is returned in row 31, we can find it with the use of the following function: @MAX(C31...F31). Pressing the ! key twice will transfer all the appropriate values to row 13.

From these summated and computed values, we may determine quite easily for each of the variables whether or not they were similar to, or quite different from, the population parameters. Table 6.3 shows a worked example using the variable of sex. All I did was place the four values, two for observed and two for expected frequencies, in the data entry table and press the ! key twice. All else was computed automatically. By simply replacing each of the appropriate variable values, I can deter-

Table 6.3

WORKSHEET FOR TESTING SAMPLE REPRESENTATIVENESS

	A	B	C	D	E	F	G	H
1			Chi Square Test of Sample Representativeness					
2								
3								
4			Data	Entry	Table			
5								
6			Attr. #1	Attr. #2	Attr. #3	Attr.#4	Attr. #5	
7								
8	Observed frequency		8	12	NA	NA	NA	
9								
10	Expected frequency		10	10	NA	NA	NA	
11								
12								
13	ChiA2	with (	1	df) =	.8333333; sig. =		.5	NA
14								
15								
16								
17			Computational		Space			Total
18								
19								
20	Observed frequency		8	12	0	0	0	20
21								
22	Expected frequency		10	10	0	0	0	20
23								
24	Cell contribut'n		.5000000	.3333333	0	0	0	.8333333
25	to IA2							
26								
27	N. of valid cells		1	1	0	0	0	2
28								
29	Degrees of freedom						=>	1
30								
31	Significance Level		.5	0	0	0	=>	.5
32								
33								
34								
35			Look-up	Table: Chi Square Statistical Sig. Level				
36								
37	Where df = 1		Where df = 2		Where df = 3		Where df = 4	
38	Comp IA2	Sig.	Comp IA2	Sig.	Comp IA2	Sig.	Comp IA2	Sig.
39								
40	0	1	0	1	0	1	0	1
41	.000628	.98	.0404	.98	.185	.98	.429	.98
42	.00393	.95	.103	.95	.352	.95	.711	.95
43	.0158	.9	.211	.9	.584	.9	1.064	.9
44	.0642	.8	.446	.8	1.005	.8	1.649	.8
45	.148	.7	.713	.7	1.424	.7	2.195	.7
46	.455	.5	1.386	.5	2.366	.5	3.357	.5
47	1.074	.3	2.408	.3	3.665	.3	4.878	.3
48	1.642	.2	3.219	.2	4.642	.2	5.989	.2
49	2.706	.1	4.605	.1	6.251	.1	7.779	.1
50	3.841	.05	5.991	.05	7.815	.05	9.488	.05
51	5.412	.02	7.824	.02	9.837	.02	11.668	.02
52	6.635	.01	9.21	.01	11.341	.01	13.277	.01
53								
54	NOTE: Press ! twice.							

mine rather easily the degree of dissimilarity, if any, between the sample statistic and population parameter. I can then use this analysis to conclude whether or not my sample selection process was biased. In this case, since there was no significant difference (e.g., chi square with (1 df) = .8333333; sig. = .5) between the sample values and its parameter, and since I checked the other variables as well and found no significant differences, I am confident that my sample is representative of the population from which it was drawn. See Table 6.3.

Now, I can go on to more interesting analysis.

Conclusion

With this article, we have begun our exploration into the world of inferential statistics. Next time we will explore how to use these ideas in the testing for significant differences on the variable of interest (e.g., such as income) for the categories of the explanatory variable (e.g., such as sex or degree). We'll look at how we may modify an earlier worksheet to include chi square analysis as part of the summary statistics of a crosstabulation and create an additional useful template.

We are rapidly coming to an end of our *Spreadsheet Series* on Data Analysis. Since we are a users group, this is a good time for me to ask for your input. What subjects would you like to see covered in future articles? Take a few moments and drop me a note on what topics, spreadsheet commands, or utilities you would like to see reviewed or examined. You may write to me at: *Jeffrey Jacques Dept. of Sociology Tallahassee, FL 32307*

Footnote

There are a number of authors who have argued that when you have one degree of freedom or when the expected cell frequencies is equal to or less than 5, then the computation of chi square should include a correction for continuity. This simply means that for each cell, find the absolute difference between the observed and expected frequencies, subtract .5, then square the remaining value. Making this slight modification to the @IF statement is not difficult. Other authors, however, argue that no such modification is necessary.

References

Blalock, Huber M. *Social Statistics: Revised Second Edition.* New York: McGraw Hill Book Company.

10382 STANFORD AVE.
SUITE E-203
GARDEN GROVE, CA 92640
PHONE: (714) 534-4216

Laing Electronics

MICROSOFT 16K CARD
$59.00

★ HARDWARE FOR THE APPLE II ★

	LIST	YOUR COST
HAYES MICROMODEM II	379.00	258.00
HAYES SMARTMODEM 300 BAUD	289.00	199.00
HAYES SMARTMODEM 1200 BAUD	699.00	498.00
NOVATION APPLE CAT II	389.00	268.00
MICROSOFT Z-80 SOFTCARD CP/M	345.00	219.00
MICROSOFT PREMIUM PACKAGE (Z-80, VIDEX 80 COL., 16K CARD)	695.00	490.00
PCPI 6 Mhz APPLI-CARD (Z-80 CP/M)	375.00	279.00
VIDEX VIDEOTERM 80 COLUMN CARD (SOFTSWITCH, AND INV. CHAR. SET)	319.00	225.00
WIZARD 80 COLUMN CARD	249.00	149.00
WIZARD BUFFERED PINTER INTERFACE (16K BUFF + GRAPHICS)	179.00	129.00
(32K BUFF + GRAPHICS)	219.00	153.00
NEC HI-RES GREEN MONITOR (JB1201)	285.00	153.00
MICRO-SCI A2 APPLE COMPATIABLE DISK DRIVE W/O CONTROLLER	345.00	229.00
MICRO-SCI A2 DRIVE WITH CONTROLLER	445.00	299.00
OKIDATA MICROLINE 92/P PRINTER	699.00	498.00
PROWRITER 8510AP PRINTER	495.00	349.00
KENSINGTON SYSTEM SAVER FAN	89.95	65.00
GIBSON LIGHT PEN/LPS II (PENTRACK)	349.00	239.00
VISTA V1200 CART. DISK SYSTEM 6.2MB	1840.00	999.00
VISTA A-800 8" DISK CONTROLLER	379.00	249.00
VISTA SOLO 5¼" DISK DRIVE	299.00	195.00
8" DRIVE SUBSYSTEM WITH CONTROLLER 2.4MB	1900.00	995.00

★ VERBATIM DISKS ★

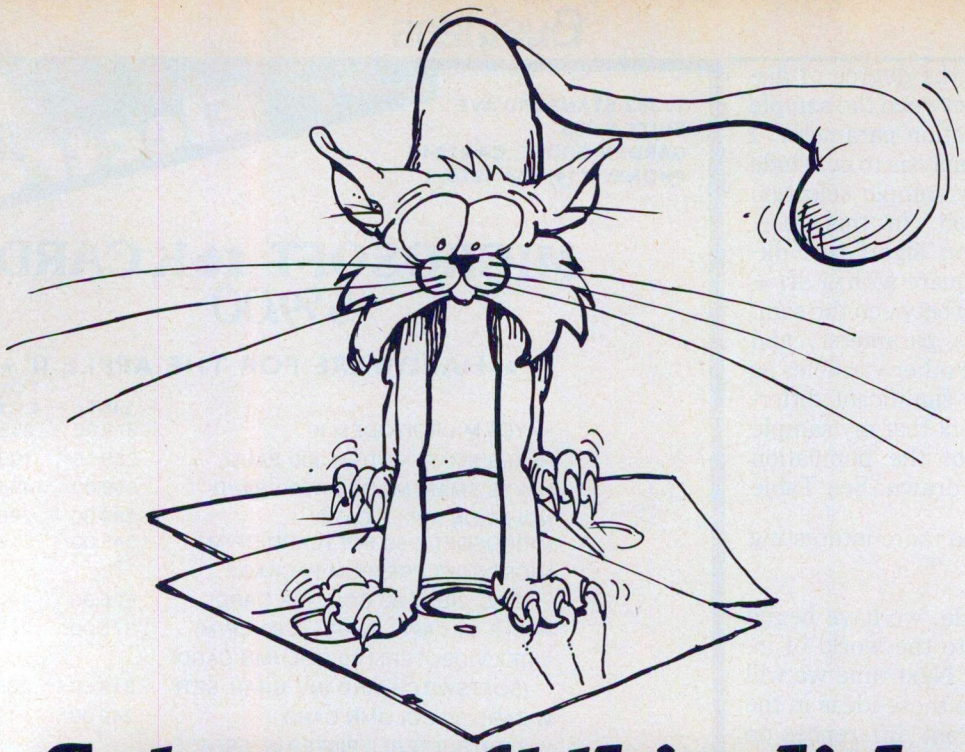
100 DATALIFE 5¼" (MD525-01-18158)	490.00	239.00
50 DATALIFE 5¼" (MD525-01-18158)	245.00	124.00

★ SOFTWARE FOR THE APPLE II ★

PFS: FILE (SOFTWARE PUB.)	125.00	76.00
PFS: REPORT (SOFTWARE PUB.)	125.00	76.00
PFS: GRAPH (SOFTWARE PUB.)	125.00	76.00
DB MASTER REG. VERSION (STONEWARE)	229.00	152.00
DB MASTER VERSION #4 (STONEWARE)	350.00	231.00
MULTIPLAN (MICROSOFT) CP/M OR DOS3.3	275.00	167.00
WORDSTAR (MICRO-PRO CP/M)	395.00	327.00
WORDSTAR WITH Z-CARD (PCPI APPLI-CARD)	495.00	425.00
SCREEN WRITER II (ON-LINE)	129.95	86.00
FORTRAN-80 (MICROSOFT CP/M)	195.00	129.00
ASCII EXPRESS PROFESSIONAL	129.95	85.00
Z-TERM PRO. (SOUTHWESTERN DATA)	149.95	98.00

If Personal Check, Allow
10 BANKING DAYS for CLEARANCE.
Prices and specifications subject to change without notice. Unless otherwise noted,
ALL PRICES are PLUS SHIPPING.
SHIPPING & HANDLING—ADD $5.00 UPS SURFACE (DISK) DRIVES $7.50—MONITORS $10.00—PRINTERS $15.00.
California residents add 6% Sales Tax.

Circle 22 on Reader Service Card



Cat on a soft thin disk.

You need software insurance.

Diskettes are fragile, and when a protected program is damaged, the results are expensive and inconvenient. If you have a backup diskette, though, you can have your Apple, IBM or compatible computer back on line within seconds . . . affordably. That's software insurance.

Copy II Plus (Apple][,][Plus, //e)

This is the most widely used backup program for the Apple. Rated as "one of the best software buys of the year" by **InCider** magazine, its simple menu puts nearly every disk command at your fingertips. The manual, with more than 70 pages, describes protection schemes, and our **Backup Book**™ lists simple instructions for backing up over 300 popular programs. A new version is now available that is easier to use and more powerful than before. Best of all, Copy II Plus is still only **$39.95.**

WildCard 2 (Apple][,][Plus, //e)

Designed by us and produced by Eastside Software, WildCard 2 is the easiest-to-use, most reliable card available. Making backups of your total load software can be as easy as pressing the button, inserting a blank disk and hitting the return key twice. WildCard 2 copies 48K, 64K and 128K software, and, unlike other cards, is always ready to go. No preloading software into the card or special, preformatted diskettes are required. Your backups can be run with or without the card in place and can be transferred to hard disks. **$139.95** complete.

Important Notice: These products are provided for the purpose of enabling you to make archival copies only. Under the Copyright Law, you, as the owner of a computer program, are entitled to make a new copy for archival purposes only, and these products will enable you to do so.

These products are supplied for no other purpose and you are not permitted to utilize them for any use, other than that specified.

Copy II PC (IBM)

This is **THE** disk backup program for the IBM PC and PC/XT that backs up almost anything. Others may make similar claims, but in reality, nothing out performs Copy II PC . . . at any price. Copy II PC even includes a disk speed check and is another "best buy" at only **$39.95.**

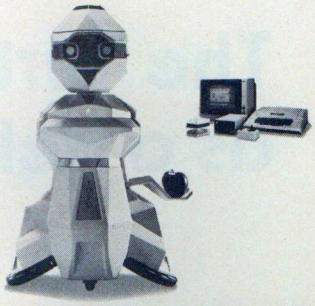
We are the backup professionals. Instead of diluting our efforts in creating a wide variety of programs, we specialize in offering the very best in backup products. So, protect your software investment, and get surefire relief from scratchy disks.

CENTRAL POINT Software, Inc.

The Backup Professionals Circle 8 on Reader Service Card

To order, call 503/244-5782, 8:00-5:30 Mon.-Fri., or send your order to: Central Point Software, 9700 SW Capitol Hwy, Suite 100, Portland, OR 97219. **Prepayment is required. Please include $2 for shipping and handling ($8** outside U.S. or Canada).

TOPO — A Tutor and Tool for Your Classroom



Harvey Barnett

THE whir of a wheel motor and the excited babble of students' voices means that TOPO, a product of Androbot Inc., is again functioning as an instructional tool at Stevens Creek School in Cupertino, CA.

Why a robot in an elementary school? To help students learn to be successful problem solvers.

A robot is a tool that can help a student to have a disciplined conversation with ideas; to develop an idea and test it using a concrete representation of an abstract idea. TOPO is a child size robot that is also an interest sustainer and confidence builder as the students struggle with their first attempts to complete a Logo procedure. The robot clearly shows the degree of correspondence between the students plan and how the robot actually traveled. Our students will need this ability to solve a problem by manipulating data if they are to compete successfully in the information age of the 21st century.

To help our students learn to be successful problem solvers, we use our robot, TOPO. TOPO is a child size robot which acts as an interest sustainer and confidence builder. Children move the robot about until he performs exactly as predicted.

TOPO is an object the children use to think with. He is a concrete representation of a powerful abstract idea — Logo. Logo is the language that makes TOPO move. It is a programming language that was developed by Seymour Paparet at MIT to help children be successful problem solvers. The problem with Logo is that it is a very abstract concept whose output is demonstrated by a turtle on a video monitor.

Piaget and Bruner espoused that children must learn to manipulate objects at the concrete level before understanding the concepts they represent at the abstract level. Getting TOPO to go in a planned pattern gives the students an opportunity to use the concept of debugging. This idea, presented by Seymour Paparet in his book *Mindstorms,* helps students understand that if they don't get a problem right the first time, they work "the bugs" out until they have achieved mastery. That is TOPO's job at Stevens Creek School. He is a concrete representation of the turtle on the screen, and as such is a powerful tool that allows us to implement Piaget's theories which stress that children should gain confidence at the concrete level before moving onto more complicated abstract levels.

Students plan a path that they want TOPO to complete. This path may be square, octagon or other geometric shapes. They write a Logo procedure that will enable TOPO to complete the planned path. If TOPO doesn't go where the child predicts, he debugs the Logo procedure that directs TOPO until it is correct.

After students have gained confidence using these first simple Logo procedures by completing a simple path or geometric shape, they go on to explore more complex procedures and movements. One of these higher level activities is to create a TOPO dance. A TOPO dance is an idea originated by David Thornburg. To perform a dance, TOPO must begin and end his movements at the same starting place and in the same orientation. When all of the students have written a dance procedure that works, the teacher can combine all of them into one super dance procedure. This super dance procedure is a good demonstration activity at a Back to School Night or an Open House.

When our students have completed these directed programming procedures and they are comfortable with the Logo commands, they are then ready to move on and explore other Logo procedures without the help of TOPO. At this point, students have developed a basic understanding of how TOPO and the screen turtle act in response to their planning and the directions they have entered into the computer.

A holiday activity for students was to write a Logo procedure that would enable TOPO to deliver a holiday greeting from the programmer to another student.

Another special activity for the robot is "TOPO Olympics." The purpose of the TOPO Olympics is to encourage students to use their Logo skills in a fun and challenging event. Another important purpose is to give public recognition to students who have mastered a computer skill in the same way we recognize a student who is a good athlete or a good speller.

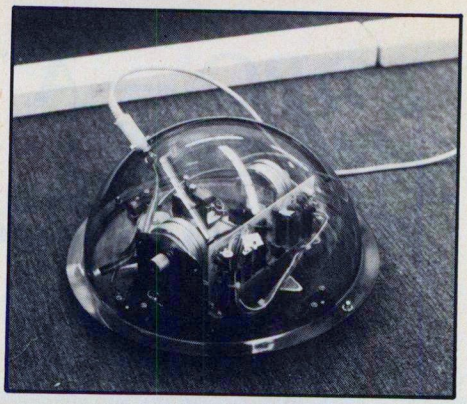
The TOPO Olympics consisted of an obstacle course of pylons that was set up in the multipurpose room with a starting and finish line. All fourth–sixth grade students were eligible to enter. Teams could consist of either two or three students. To win, a team had to write Logo procedures that would direct TOPO through the course faster than the other teams.

Twelve teams entered in the first TOPO Olympics. The top three winning teams represented all levels of student achievement from learning disabled to gifted. What made for a winning team was the ability of the team to plan and solve the problem together.

By working together, the team members met one of the important

cont'd on p. 67

The Turtle as a Helpful Robot



Nola Sheffer

WHAT do you picture when the word "robot" is mentioned? R2D2? CP3O? A gimmicky toy? Let me introduce you to the robotic turtle, which resembles neither a toy, nor something out of a science fiction movie. Thanks to its size and speed, it can be a very good device for teaching concepts in the elementary curriculum and it is not at all intimidating to adults.

The first robotic turtles were developed in England by the cybernetician Robert Gray. The MIT Logo Project used robotic turtles, as well as the now-famous screen "turtle." Because we use Logo in our school, and believe strongly in the use of hands-on concrete materials for young children, we purchased a robotic turtle. A few months later we purchased our second Tasman, and are currently trying out the Tot with the idea of purchasing it as our third turtle.

Typical of remarks made by children when the turtle is first presented are:

"That's a turtle?!"

"Its shell looks like one."

"I like the wheels on it — it looks like a racing car."

While neither turtle could really be called "cuddley," each seems to have appeal — a useful attribute for an educational aid.

At first we used the turtle to make designs and geometic shapes, the way one uses the screen turtle. However, we soon raised the pen, took away the paper, and began to let the turtle "explore." In so doing, we broadened the use of the turtle, and linked it to several different subject areas and skills. We have found it useful in reinforcing language arts skills, such as sequencing and story-telling, and as a motivation for writing. It ties in very nicely with directionality and mapping, and is a natural for estimation and introduction to angles.

Both the Tasman Turtle, and its smaller offspring, The Tot, come with software that allow it to respond to Logo or BASIC commands. If using Logo, you "read" or "load" Logo into your machine, and then load in the Logo primitives written to interface with the robotic turtle. For example, to move the turtle forward 40 turtle steps, you type TFD 40. (If you type FD 40, the screenturtle moves, just as it would if the robotic turtle were not connected.)[1]

In using the turtle with kindergarten and first-grade children, we use a single-keystroke Logo program, and I usually write it so that pushing "F" moves the turtle forward an inch or two at a time, and pushing "B" moves the turtle back that same distance. (For the Tasman, I chose to make "F" result in a move of 20 turtle steps. Since the Tot has smaller steps, I made "F" produce a move of 50 turtle steps. I have found that a 30 degree turn is convenient for "R" and "L." (See inset for sample "Instant" Program.)

```
(MIT LOGO FOR THE APPLE)

TO INSTANT.TURTLE
DRAW
DIRECT
END

TO DIRECT
GCOLL
MAKE "KEY READCHARACTER
IF :KEY = "F TFD 50
IF :KEY = "B TBK 50
IF :KEY = "R TRT 30
IF :KEY = "L TLT 30
DIRECT
END
```

```
(APPLE LOGO)

TO INSTANT.TURTLE
CLEARSCREEN
DIRECT
END

TO DIRECT
RECYCLE
MAKE "KEY READCHAR
IF :KEY = "F [FD 50]
IF :KEY = "B [BK 50]
IF :KEY = "R [RT 30]
IF :KEY = "L [LT 30]
DIRECT
END
```

Placing small pieces of masking tape on the four keys used reduces the searching needed to find the right letter. Most kindergarten children do not know letter names or sounds, yet this does not seem to be a problem. Since there is no "right" or "wrong" in this activity, the child merely tries another key to get the desired result, and soon the children know what each key does.

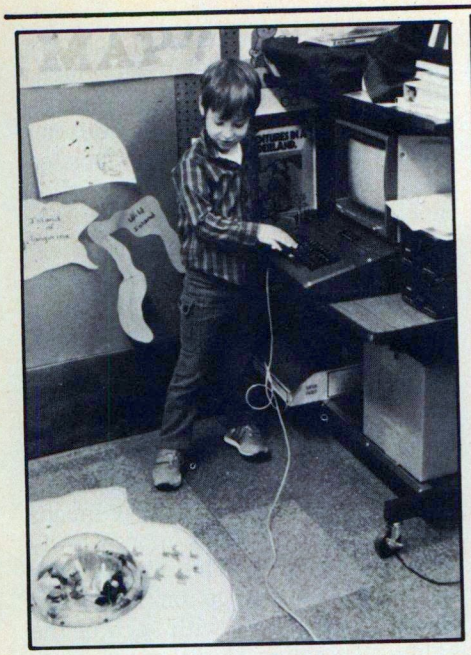
Using these four keys, the children first experiment with moving the turtle around, observing the response of the turtle. Soon they are ready to put down blocks to make paths for the turtle, or to "park" it under a table. Most classrooms have plenty of objects of interest which the turtle can visit.

"That's not what I thought it would do."

"Oh, it got back where it started."

These are typical remarks of students directing the turtle. In all of this they are getting practice in directionality, and observing opposites in action. They also experience directionality as relational, since the turtle's left is not necessarily the same as theirs.

[1]MIT Logo for the Apple. (In Apple Logo, the commands for screen and floor turtles are the same.)



The tiled floors of classrooms are a handy tool in estimating distances needed to move the turtle from one location to another. The software with one of the turtles is so written that there are approximately 110 turtle steps to one eight-inch tile. This results in some good mental addition practice for the third-graders using this turtle.

The ability of the user to be in charge is one of the central themes of the philosophy of Logo. For me it is

made especially powerful when using the turtle. Because it is slow and plodding, one is not distracted by flashy effects. Yet students find it fun to direct — girls as well as boys. When working with the turtle, it is easier for me to concentrate on the process of Logo, of using the turtle as an "object to think with," and not be pushing children to get to the next stage in programming.

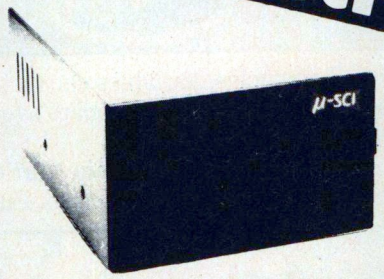
The importance of learning as a process, and using the turtle as an "object to think with" are discussed in *Mindstorms,* a book by Seymour Papert which delineates the philosophy of Logo. Papert also says that it is important for learning to be syntonic — to be meaningful in a personal way. A part of this is physical syntonicity. Students "play turtle," that is, act out the turtle commands of forward, back, left, and right when they are first learning movements. Then when they are uncertain how they want the turtle to move, they return to acting out moves with their own bodies, before typing the commands. Thus the turtle has a personal connection in a way that would be difficult for any image on the screen.

All the activities mentioned so far are ones I have used with primary school children. But the use of the

INTERESTING OR UNUSUAL APPLICATION OF YOUR APPLE?

Call -A.P.P.L.E. wants to bring its readers the most interesting and unusual applications of Apple computers. If you think the way you use your Apple would be interesting to the rest of the club, please write and tell us about it. It could lead to a story published in Call -A.P.P.L.E. To let us know; A.P.P.L.E., Applications, 21246 68th Ave. S., Kent, WA. 98032.

Laing Electronics



μ-SCI
MICRO-SCI
A2 DRIVE

$229

BUY THE FIRST AND THE BEST APPLE COMPATIBLE DRIVE!

A2 DRIVE WITH CONTROLLER $309
(FREE SHIPPING IN USA OR CANADA. A2 MODELS ONLY.)

A40 (40 TRACK DRIVE W/O CONTROLLER) $284
A70 (70 TRACK DRIVE W/O CONTROLLER) $365
— CONTROLLER FOR A40 OR A70 DRIVES $ 79
A3 APPLE III DIRECT REPLACEMENT $299

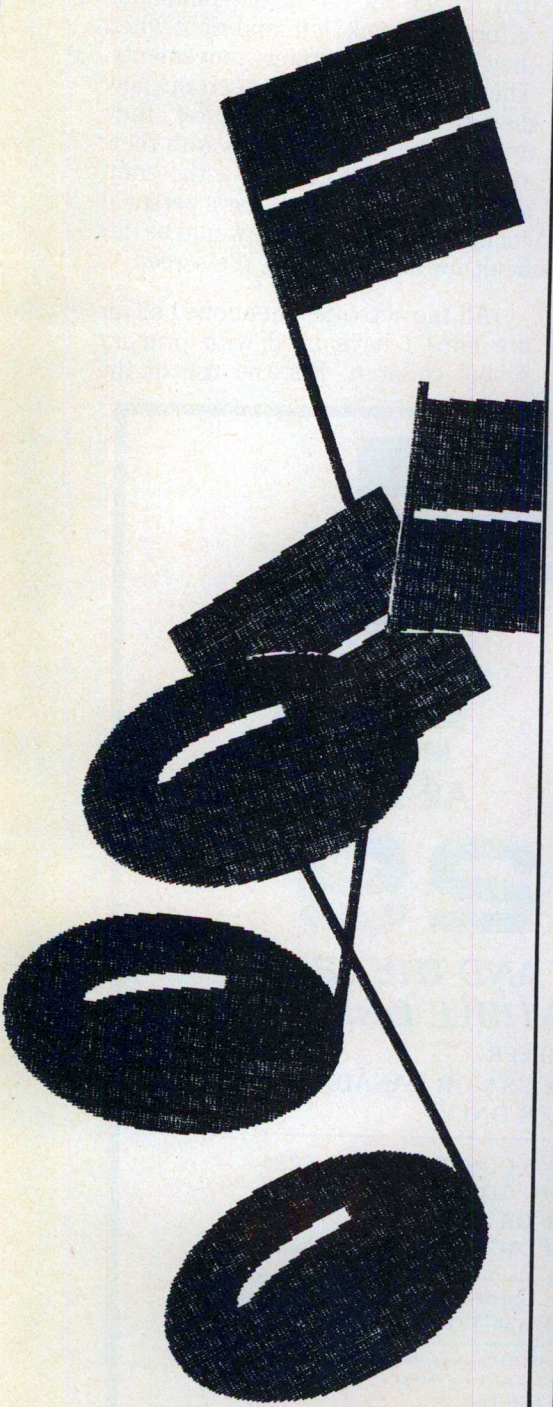
10382 Stanford Ave.,E-203, Garden Grove, Cal., 92640
Phone (714) 534-4216

Check or Money Order. — NO CREDIT CARDS! Check's Allow 10 BANKING DAYS FOR CLEARANCE. Prices and specifications subject to change without notice. Add $7.50 for UPS surface. Cal. Res. add 6% Sales Tax.

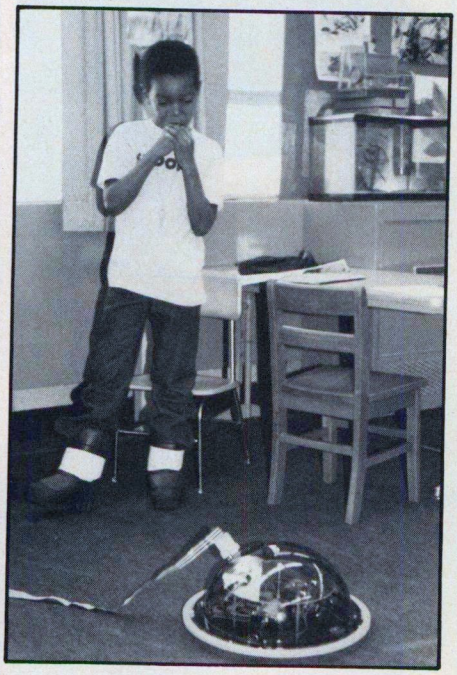
Circle 18 on Reader Service Card

Next Month in Call A.P.P.L.E. watch for our issue on

Computers and Music!



turtle is not restricted to young children, it is appropriate to all ages. The turtle has touch sensors (a bumper-like ring on the Tasman; a sensitive rim on the Tot) which allow one to program the turtle to move around a space, moving away from objects that it bumps. Such a program provides a nice introduction to recursion and conditionals. Then, in trying to find just the right amount of turn and backing up needed to get



around obstacles, variables can be introduced as a convenient tool to avoid a lot of editing.

As mentioned before, the turtles are not limited to use with Logo. Furthermore, they have other features, such as speech synthesizers, which I have not used. I feel there is a great deal of potential for these turtles in a variety of areas. Harvard Associates, the U.S. distributors of the turtles, sell many to universities for the study of robotics.

The Tasman turtles are quite expensive at nearly $1000. The Tot, at $300 ($395 if you need a serial interface) begins to be at an approachable price for schools. Since the Tot is serial, its "tail" or power cord can be much longer, allowing for a greater distance to be travelled from the computer. Being serial also means that the Tot responds more slowly than the Tasman. The Tasman has four lights which go on when the bumper is touched, with each of the lights responding to different points on the rim; the Tasman also has a horn. The Tot does not have these features. While the Tasman also has some other features which the Tot does not, the Tot is the better buy for most schools, considering the price differential.

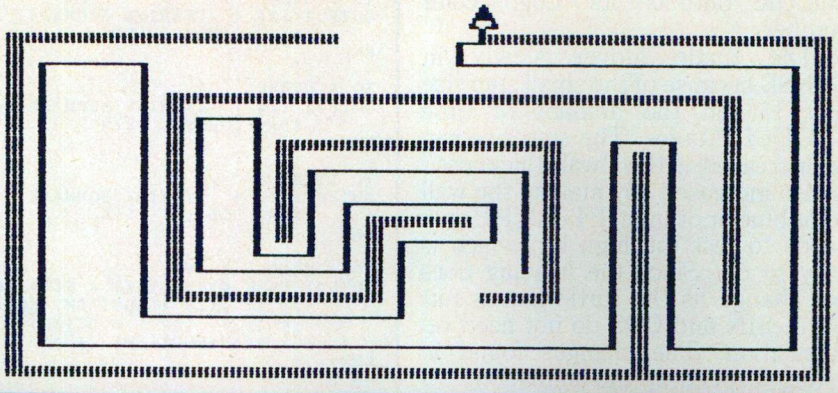
Robots will not take over the teaching or the learning tasks in a classroom, yet robots are a tool for all educators who believe strongly in hands- on, concrete materials.

Melissa's Story

Once there was an old, old turtle. He was so old that he wanted somebody to play with. He couldn't find anybody to play with, so he wandered in the park, and then he wandered in the woods, then he wandered back to town and sat down in his old chair.

```
TO MELISSA
TFD 78
EYESON
TRT 8
EYESOFF
TRT 78
TFD 78
TRT 78
TFD 70
TRT 89
EYESON
TOOT
TLT 90
TRT 9
TFD 9
TOOT
TRT 9
TFD 9
TOOT
END
```

Touch Sensors for Screen Turtles



Jeff Sandys

WHEN it was first used in a classroom in 1968, Logo did not have turtle graphics. Floor turtles were added to Logo as an object to think with. Controlling these robots with the computer gave the students a sense of power. Named for their domed shaped cover, floor turtles were used simply because graphics terminals were too expensive and rare. Thanks to the Apple Computer, screen turtles far outnumber floor turtles today.

Floor turtles have one feature not commonly found on screen turtles. Switches on the front and sides provide floor turtles with a sense of touch. We can add touch sensors to our screen turtle by reading the hi-res screen. This article will explore screen reading techniques and use touch sensors in a maze solving algorithm. Also presented is a maze drawing utility.

The trick of reading the hi-res screen is converting Logo coordinates into a memory address. The Apple hi-res screen is folded into the memory in a peculiar way. Briefly stated, the 192 horizontial lines are divided into 3 groups of 8 rows per group, of 8 lines per row, of 40 bytes per line, of 7 dots per byte; groups are spaced 40 bytes apart, rows within groups are spaced 128 bytes apart, and lines within rows are spaced 1024 bytes apart. If this seems confusing, don't worry, you are not alone. Read your Apple Reference Manual or one of the many books on Apple graphics for more information. The SCRNADR procedure is an algorithm that will unfold the screen. Just give it a column and line and it will output the memory address.

The memory location found with SCRNADR contains seven dots. BITP is used to examine any bit in a byte. Quotient and remainder shift out the requested bit for testing. The helper procedure EXP raises a number to an integer power. The procedure SCRNP combines these procedures to test any hi-res screen dot.

SCRNP uses the Applesoft hi-res coordinate system. Columns are numbered left to right, 0 to 279. Lines are numbered 0 to 191 downward. Logo uses the center of the screen as the origin. Logo's columns are numbered from minus 140 to plus 139. The vertical position is positive upward and proportioned with the SCRUNCH ratio. The LINE and COLUMN procedures convert Logo coordinates to Applesoft coordinates.

Now we can construct a procedure to read the screen using Logo coordinates. TOUCHP outputs !dqTRUE if the specified dot is on. FTOUCHP RTOUCHP, and LTOUCHP test the screen near the turtle. Position and heading are used to calculate the location of the turtle's corners.

The procedure SENSE in listing 1 will test the touch sensor. First draw a box with:

```
REPEAT 4 [FD 40 RT 90]
```

then place the turtle in it with:

```
PU SETPOS [20 15]
```

Enter SENSE to start the animation and use [Ctrl-G] to stop.

Listing 1
SENSOR.LOGO

```
TO SENSE
IF FTOUCHP [RT 160 + RANDOM 40]
FORWARD 1 SENCE
END

TO FTOUCHP
OUTPUT TOUCHP (XCOR + 12 * SIN
      HEADING) (YCOR + 12 * COS
      HEADING)
END

TO RTOUCHP
OUTPUT TOUCHP (XCOR + 8 * COS
      HEADING) (YCOR - 8 * SIN
      HEADING)
END

TO LTOUCHP
OUTPUT TOUCHP (XCOR - 8 * COS
      HEADING) (YCOR + 8 * SIN
      HEADING)
END

TO BITP :BIT :BYTE
OUTPUT 1 = REMAINDER (QUOTIENT
      :BYTE EXP 2 :BIT) 2
END

TO EXP :N :E
IF :E = 0 [OUTPUT 1] [OUTPUT :N
      * EXP :N :E - 1]
END

TO LINE :Y
OUTPUT ROUND SCRUNCH * (120 -
      :Y)
END

TO COLUMN :X
OUTPUT 140 + :X
END

TO SCRNP :X :Y
OUTPUT BITP REMAINDER :X 7 .EXAMINE
      SCRNADR :X :Y
END

TO TOUCHP :X :Y
OUTPUT SCRNP COLUMN :X LINE :Y
END

TO SCRNADR :X :Y
OUTPUT (SUM 8192 (1024 * REMAINDER
    :Y 8) (128 * QUOTIENT REMAINDER
    :Y 64 8) (40 * QUOTIENT
    :Y 64) (QUOTIENT :X 7))
END

TO COLOR :X :Y
MAKE "PIXEL TOUCHP :X :Y
MAKE "ADJNT OR TOUCHP :X + 1 :Y
      TOUCHP :X - 1 :Y
TEST :PIXEL = :ADJNT
IFTRUE [IF :PIXEL [OUTPUT 1] [OUTPUT
      0]]
TEST BITP 7 .EXAMINE SCRNADR COLUMN
      :X LINE :Y
IFFALSE [MAKE "HI 2] IFTRUE [MAKE
      "HI 4]
IF :ADJNT [OUTPUT :HI + REMAINDER
      :X 2]
OUTPUT :HI + REMAINDER :X + 1
      2
END
```

TOUCHP can also be used to find the color of a dot. The color is determined by three factors: the state of the high bit, the state of the adjacent dots, and if the dot is in an even or odd column. The three IF statements in the COLOR procedure sort these factors. Given the dot's coordinates,

COLOR outputs its Logo color number.

The turtle moves slowly in SENSE because of the small step size (FD 1) and the amount of time FTOUCHP takes. The step size can be increased if the "wall thickness" is also increased. By making the wall color blue or orange, FTOUCHP only needs to test the high bit, which is easy to do. Since the heading does not change as the turtle moves forward, SIN and COS do not need recalculation. The changes found in listing 2 speed up the animation.

Listing 2
HI.TOUCH.LOGO

```
TO MOVE
IF FTOUCHP [STOP]
FORWARD 3.5 MOVE
END

TO TURN :N
RIGHT :N
MAKE "SIN 10 * SIN HEADING
MAKE "COS 10 * COS HEADING
END

TO LFOLLOW
IF OR LTOUCHP = "FALSE FTOUCHP
     = "TRUE [STOP]
FORWARD 3.5 LFOLLOW
END

TO RFOLLOW
IF OR RTOUCHP = "FALSE FTOUCHP
     = "TRUE [STOP]
FORWARD 3.5 RFOLLOW
END
```


BYTES & PIECES **PRICE RIOT** / Ask For Free Price List

$1.00 credit for phone orders over $100.00

	LIST	SPECIAL
Bookends	125.00	90.46
Computer Baseball	39.95	28.56
Crossword Magic	49.95	35.96
D.B. Master 4.0	350.00	252.96
Ess. Data Dup. (EDD)	79.95	58.16
Flight Simulator II	49.95	36.96
Fontrix	75.00	54.46
G.P.L.E.	49.95	35.46
Graphics Magician	59.95	43.36
Home Accountant	74.95	53.46
Homeword	69.95	49.96
Incredible Jack	179.00	130.46
Koala Pad	124.95	90.46
Krell Logo	89.96	75.46
Krell SAT	299.95	252.96
New Step by Step	89.95	64.56
PFS File (II+ or IIe)	125.00	90.46
PFS Write	125.00	90.46
Planetfall	49.95	35.46
Sensible Speller	125.00	90.46
16 K Card (Microsoft)	99.95	75.96
Think Tank (II+ or IIe)	150.00	108.96
Trains	39.95	28.56
Wildcard II	139.95	110.19
Wildcard +	169.95	133.56

Ask for our educational software catalog

C.O.D. • M.O. • Cert. Checks • School P.O. • VISA • MasterCard • Checks Allow 2 Weeks • N.Y.S. Res. Add Sales Tax • US Orders Under $150. Add $2.00 P & H • All Canadian, US Funds $3.00 P & H. • Foreign, Charges Only, Min. P & H $7.00

Bytes & Pieces, Box 525, Dept. C,
East Setauket, N.Y. 11733, (516) 751-2535

Circle 13 on Reader Service Card


```
TO FTOUCHP
OUTPUT 128 < .EXAMINE SCRNADR
     X + :SIN Y - :COS
END

TO RTOUCHP
OUTPUT 128 < .EXAMINE SCRNADR
     X + :COS Y + :SIN
END

TO LTOUCHP
OUTPUT 128 < .EXAMINE SCRNADR
     X - :COS Y - :SIN
END

TO SCRNADR :X :Y
OUTPUT (SUM 8192 (1024 * REMAINDER
     :Y 8) (128 * QUOTIENT REMAINDER
     :Y 64 8) (40 * QUOTIENT
     :Y 64) (QUOTIENT :X 7))
END

TO Y
OUTPUT .EXAMINE 926
END

TO X
OUTPUT SUM .EXAMINE 924 256 *
     .EXAMINE 925
END
```

The conversion of Logo coordinates can also be eliminated with a little trick. When Apple Logo draws the turtle, it also leaves the turtle's line and column in page 3 memory. The procedures X and Y examine these locations and output the turtle's Applesoft coordinates. This saves some time and provides for an accurate turtle position.

The new touch procedures are used to implement MOVE, TURN, RFOLLOW and LFOLLOW. MOVE runs the turtle until it hits a wall. RFOLLOW and LFOLLOW track an adjacent wall. TURN is a right turn that also sets the SIN and COS variables.

Walls need to be drawn in order to use MOVE, RFOLLOW and LFOLLOW. The drawing utility in listing 3 will make drawings and save the instructions to a disk. To start the program, enter DRAW. The "@" prompt indicates that the program is running. The commands that are available are:

I	move up
J	move left
K	move right
M	move down
L	mark the lower-left wall corner
P	put a wall from L to here
O	mark the maze opening
E	erase the drawing instructions
S	save the drawing instructions
Q	quit the drawing program

Listing 3
DRAWING.LOGO

```
TO J
SETX XCOR - 7
END

TO K
SETX XCOR + 7
END

TO I
SETY YCOR + 7
END

TO M
SETY YCOR - 7
END

TO Q
THROW "TOPLEVEL
END

TO E
MAKE ".DWG [SETPC 4] DRAW
END

TO O
MAKE ".DWG (SE "MAKE ""O YCOR
     :.DWG)
END

TO L
MAKE "L SE XCOR YCOR
SETH 45 SETPC 1 PD DOT :L SETPC
     4 PU
END

TO P
IF HEADING = 225 [TYPE CHAR 7
     STOP] [SETH 225]
MAKE "POS POS HT TEST XCOR = FIRST
     :L
IFTRUE [MAKE "P (SE "VLIN LAST
     :L YCOR - 1 XCOR)]
IFFALSE [MAKE "P (SE "HLIN FIRST
     :L XCOR - 3 YCOR)]
MAKE ".DWG SE :.DWG :P RUN :P
     SETPOS :POS ST
END

TO S
SETH 0 (TYPE [ENTER MAZE NAME:]
     CHAR 32)
MAKE "FILE WORD LAST RL ".DWG
     (PRINT "SAVING :FILE)
MAKE ".DWG (SE :.DWG "SETPOS (LIST
     SE XCOR YCOR))
PPROP "VLIN "PROCPKG ".DWG PPROP
     "HLIN "PROCPKG ".DWG
PPROP ".DWG "VALPKG ".DWG SAVE
     :FILE ".DWG DRAW
END

TO DRAW
IF NOT NAMEP ".DWG [E]
CLEARTEXT CS PU ST SETPOS [1 0]
     SETH 225 RUN :.DWG MAIN
END

TO MAIN
SETCURSOR [0 20] TYPE "@ MAKE
     "RC RC SETCURSOR [0 20]
     PR []
IF MEMBERP :RC [E I J K L M O
     P Q S] [RUN SE :RC] [TYPE
     CHAR 7]
MAIN
END

TO HLIN :X0 :X1 :Y
REPEAT 5 [SETPOS SE :X0 :Y PD
     SETPOS SE :X1 :Y PU MAKE
     "Y :Y + 1]
END

TO VLIN :Y0 :Y1 :X
REPEAT 3 [SETPOS SE :X :Y0 PD
     SETPOS SE :X :Y1 PU MAKE
     "X :X + 2]
END
```

There are some rules to follow to insure a successful maze drawing. After pressing "L," only move up with "I," or right with "K." This is to make "P" put the walls within byte boundaries. At the top of the drawing, leave some space and mark the opening in the maze with "O." Before saving the drawing, move the turtle to the maze starting position. The saved drawing doesn't need the drawing program. Load the drawing

(with the .DWG extension) and enter:

RUN :.DWG

At last the touch sensors can be put to use. The example in listing 4 executes the Pledge algorithm. At age 12 John Pledge developed this maze solving procedure for the turtle. By keeping track of the total turn, the turtle can find its way out of any fair maze. Here are the steps for the procedure:

1) Set the total turn to zero.
2) Move forward until an obstacle is encountered.
3) Turn left until the obstacle is on the right, while keeping track of the total turn.
4) Follow the obstacle, keeping it on the right until the total turn is zero.
5) Go to step 2 until escaped.

Step one is performed by the START procedure. The procedure PLEDGE does steps 2 and 3 while FOLLOW.RIGHT does steps 4 and 5. The TURN procedure has been modified to record the total turn. ESCAPED terminates the program with a message. The other procedures used here are from listing 2.

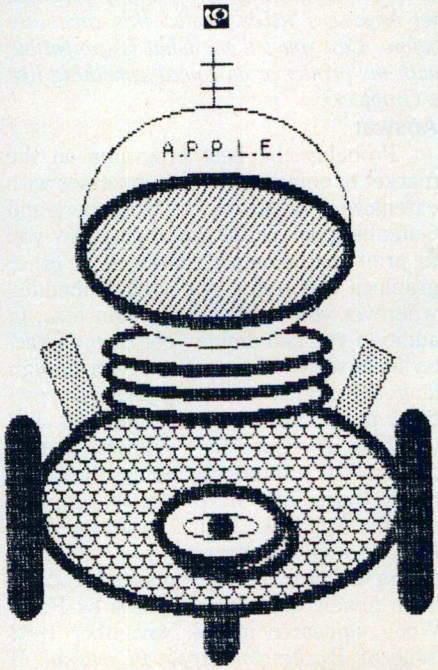
Listing 4
PLEDGE.LOGO

```
TO START
( TYPE [ENTER MAZE NAME:] CHAR
      32 ) LOAD WORD LAST RL ".DWG
CS PU HT RUN :.DWG SETH 0 SETPC
      1 PD ST ERALL ".DWG
MAKE "TOTAL.TURN 0 TURN 0 PLEDGE
END

TO PLEDGE
IF YCOR > :0 [ESCAPED]
MOVE TURN -90 FOLLOW.RIGHT PLEDGE
END

TO FOLLOW.RIGHT
IF YCOR > :0 [ESCAPED]
IF :TOTAL.TURN = 0 [STOP] [RFOLLOW]
IF FTOUCHP [TURN -90] [FD 3.5
      TURN 90 FD 7]
FOLLOW.RIGHT
END

TO ESCAPED
PR [I ESCAPED !!!] REPEAT 4 [TYPE
      CHAR 7] THROW "TOPLEVEL
END

TO MOVE
IF FTOUCHP [STOP]
FORWARD 3.5 MOVE
END

TO TURN :N
RIGHT :N
MAKE "TOTAL.TURN :N + :TOTAL.TURN
MAKE "SIN 8 * SIN HEADING MAKE
      "COS 8 * COS HEADING
END

TO RFOLLOW
IF OR RTOUCHP = "FALSE FTOUCHP
      = "TRUE [STOP]
FORWARD 3.5 RFOLLOW
END
```

```
TO RTOUCHP
OUTPUT 128 < .EXAMINE SCRNADR
      X + :COS Y + :SIN
END

TO FTOUCHP
OUTPUT 128 < .EXAMINE SCRNADR
      X + :SIN Y - :COS
END

TO SCRNADR :X :Y
OUTPUT (SUM 8192 (1024 * REMAINDER
      :Y 8) (128 * QUOTIENT REMAINDER
      :Y 64 8) (40 * QUOTIENT
      :Y 64) (QUOTIENT :X 7))
END

TO X
OUTPUT SUM .EXAMINE 924 256 *
      .EXAMINE 925
END

TO Y
OUTPUT .EXAMINE 926
END
```

Because the maze walls are perpendicular, 90 degree turns can be used. Odd shaped obstacles would require a different follow routine. For more information on the Pledge algorithm, see page 177 of *Turtle Geometry* by Abelson and diSessa, or page 18 of the February 1984 issue of *Scientific American.*

I hope you have found some useful procedures here. BITP, EXP and COLOR have many possible applications. The drawing utility can be jazzed up with more functions. The Pledge algorithm is just a start for maze solving. Routines can be added that learn the path and then try to optimize it. Finally I would like to apologize to MIT Logo users; many of the routines given here will only work with Apple Logo. I just got a copy of Terrapin Logo and in the future will try for more compatibilty.



cont'd from p. 61

BASIC language the robot understands (for example, translating Figure 1 into Figure 2), and to download the robot task to the robot via an RS-232 serial communications interface. Once the downloading process is completed, RB5X may be disconnected from the Apple to allow it to execute the robot task.

Clearly, RCL is an easy system to use in programming RB5X. It is the governance of Savvy, however, that allows RCL to be not only simple but also very flexible. Savvy is a personal language software system developed by Excalibur Technologies Corporation. Savvy incorporates a computer operating system, utilities (for example, diskette copying capability), and an indefinitely extensible computer language.

RCL is actually an application language that exists within Savvy. RCL users have access to all Savvy features and functions for system operation, and for robot task building and alteration (though some Savvy operations do not apply to robot programming).

One feature of Savvy that is pargoals we set for our computer program. They used their Logo skills and they were able to plan and work at a problem until it was solved successfully by having TOPO cross the finish line.

TOPO is just a means to an end. After he has demonstrated to students their beginning Logo efforts, he sits quietly in the corner while the students work at the computers creating even more interesting Logo procedures with the assurance that their first tries might not always be what they wanted, but that by debugging and working toward the idea they have predicted, they will succeed. Students not only see themselves as successful problem solvers, they also acquire the skills they need to do the job.

A.P.P.L.E. Doctor

Bruce F. Field

I have an update to an answer that appeared in the February 84 A.P.P.L.E. Doctor column. The questioner wanted to know what communications programs would work with his Apple Super-serial card and acoustic modem. The latest versions of ASCII Express II, ASCII Express "The Professional," and P-Term (The Professional) as well as VisiTerm and Datalink all work with the Super-serial card.

In the January 84 A.P.P.L.E. Doctor column, I replied that Apple Computer Inc. would upgrade Apple //e's with rev. A motherboards to rev. B if the user could prove that they had an Extended 80 Column Card (the one with the extra 64K of memory). (The rev. B board allows double high resolution graphics 560x192 with the Extended 80 Column Card.) At that time, I suggested that the 80 Column Card might have to be Apple's and not another brand. Charlie Symmons recently received a reply from Apple which he passed along to me stating that *any* Extended 80 Column Card either made by Apple or another vendor will satisfy the above requirement.

A friend of mine ran across an interesting problem the other day that may serve as a warning to others. He was working at his computer when there was a very short power outage. Fortunately the duration was so short that his program and data in memory were still okay. After saving the program to disk, he was unable to catalog the disk or perform any disk operation without the dreaded I/O ERROR. The directory on the disk was apparently scrambled. What we think happened was this. The RAM memory chips in the Apple will more or less retain their data for a little while after the power is turned off. This data retention time ranges from a few milliseconds to sometimes as much as a second and depends on the characteristics of the individual chips. Apparently the chips containing DOS were somewhat more sensitive to the power loss and scrambled DOS while the program remained intact. When the program was saved to disk, it was saved improperly and the directory was destroyed. Although this sounds like a freak occurrence it actually happened to this fellow twice, and it wasn't until the second time that it occurred to him what was happening. MORAL: If you have even a brief power outage and you want to try to save your program or data to disk, use an expendable disk that has been backed up. (Of course we have backup copies of *all* our disks don't we?)

Question

My wife is a high school math teacher (with little interest in computers. Can you believe it?) and I try to help her by using the computer. This year she is teaching Geometry and I've met the challenge and lost. I can't figure a way to print geometric figures, with labels, near the applicable question. I am fairly convinced that I need a capability that combines graphics with different text sizes and styles on the same screen. Most graphics programs that I've read about or seen have limited text positioning capability or the text alpha-numerics are too large for my purposes. Can you help? By the way, I have not bought a graphics dump program yet because I never needed this capability before. Can you tell me what is compatible with my printer or do I need something like a Grappler?

Answer

Probably the best programs on the market to combine a word processor with extensive graphics are the Gutenberg and Gutenberg Jr. These programs allow you to print mathematical symbols, or other graphics that you can design, imbedded wherever you want them in your text. In addition you can define your own character fonts with addition symbols for foreign languages. These are exceptionally powerful programs for combining text and graphics. The Gutenberg supports several dot matrix printers including the NEC 8023. Gutenberg Jr. presently supports only the Apple Dot Matrix Printer and the *IIe*, although by the time you read this, others will probably be supported. A complete review of these programs by P. K. Wong, appeared in the November 1983 issue of *Washington Apple Pi Journal.* If the programs cannot be found locally, they can be ordered from Micromation, Ltd., 1 Yorkdale Road, Suite 406, Toronto, Ontario, Canada M6A 3A1. Gutenberg costs $325 and the Jr. $85.

At the risk of being labeled a heretic, isn't it also possible to use any of the graphics programs to draw your graphics on the screen (with labels), dump them to a printer, and then cut them out and paste them where they are supposed to go in the text? Seriously this is not such a bad solution. Trying to position text and graphics exactly where you want them on the paper can be a time consuming process. Of course if you want to get ridiculous about it, sell your Apple][and get the new Apple Macintosh which is made for mixing text and graphics.

The high-res dump program Zoom Graphix (from Phoenix Software) can be used to dump graphics to your NEC 8023 printer using the standard interface card you already have. Zoom Graphix and like programs offer more capability for scaling, windowing (Windowing — Is that a word? Am I guilty of creating more computer jargon? Naa, Microsoft uses it.), and rotating the image than the interface cards with the built in dump programs like the Grappler. The Grappler is just more convenient.

Being in Germany, if you find it hard to keep up on new programs you might want to get a copy of one of the software books that list most of the software available for the Apple. *The Blue Book,* published by WIDL Video Publications, Chicago IL, is one of the most comprehensive. *The Book of Apple Software 1984* by The Book Company and Addison-Wesley contains fewer entries but provides independent reviews of all the programs listed. I think the latter is one of the better software reference books.

Question

The "single wire" shift key modification, which connects bar 24 on the keyboard encoder card with one of the pins on the Apple][+ game port seems to have disabled input from PDL(1). I have confirmed continuity through all pins of the shift key modification, and I am certain that everything is

seated properly. When the shift key modification is removed, the game port works as it always had. Is this the price I have to pay for achieving true shift key operation?

Answer

No, you apparently have a wiring error. The wire from the keyboard should go to pin 4 of the game I/O connector. The game I/O connector is at the rear of the computer near the right hand side as you face the computer (it's labeled on the motherboard). The pins are numbered starting on the right hand side at the end nearest you. Pin 4 therefore is the fourth pin back on the right hand row of pins. If you have everything connected correctly, the only feature you lose is the third push-button input (usually called PB2, but sometimes PB3). This should in no way affect your paddle readings.

For more information see the article "Curing a Shiftless Apple" by Dan Paymar in the May 1982 issue of *Call -A.P.P.L.E.*

Question

I would like to use my computer and NEC 8023 printer like a typewriter for filling out forms; this would make it easy to determine spacing and font size for developing programs to automatically complete certain forms. Is it possible to make the printer respond immediately to the striking of a key or is the return key the only way to send info to the printer?

Answer

There are two problems here that have to be fixed. First, the printer generally saves up the characters sent to it in a buffer and only prints the entire line when a carriage return character is received. This can be fixed by putting the printer in the incremental print mode by sending the "[" character CHR$(91). The printer will print each character as it receives it. Second, we need to write a short program that will send each character to the printer as soon as you type it. The following Applesoft program will do this.

```
100  PRINT CHR$(4)"PR#1"
     :REM TURN ON PRINTER IN
     SLOT 1
110  PRINT CHR$(91);
     :REM SET PRINTER TO
     INCREMENTAL MODE
120  GET A$ : PRINT A$;
     : GOTO 120
```

The only way to stop this program is to press RESET. This program will also work with the C. Itoh 8510 Prowriter, but *not* the Apple DMP.

Question

*In using Applewriter II's embedded command, .tl**page #**, to number all pages except the first one, the header on each page is printed one space lower on the page than on the preceding one. After several pages, the header appears halfway down the page. What is causing this? I have used two different disks for the Applewriter program, so the problem apparently is not there.*

Answer

Funny you should ask, I just ran across this problem while working on this column. The problem is that when you print the title (or header) it is printing a full 80 columns which then causes your printer (or printer interface card) to insert an extra line feed unbeknownst to Applewriter. (In my case it was the printer.) If you cannot stop this extra line feed, then use the left and right margin settings to limit the line length to less than 80 columns. Remember if the header is typed directly into the default print value menu, then the header is generated according to the margin settings in the default print value menu, so in this case changing the margins in your text (i.e. .lm, .rm) will not help.

Often problems with paging on Applewriter are related to the setting of the top and bottom margins. First and foremost the page interval (PI) or number of lines per page should be set to 66 assuming you are using 11" long paper at six lines per inch. The top margin ™ is the number of blank lines printed after the header (or top line, which in this case is the page number) is printed. Skipping the body of text for the moment, we come to the bottom of the page. The bottom margin is the number of blank lines printed between the last line of text and a footer (or bottom line) if one exists. PL or printed lines controls the total number of lines printed on the page including the header, top margin, body of text, bottom margin, and footer. Thus the body of the text is determined by PL the printed lines minus the top margin, minus the bottom margin, minus one for a header if it exists, minus one for a footer if it exists.

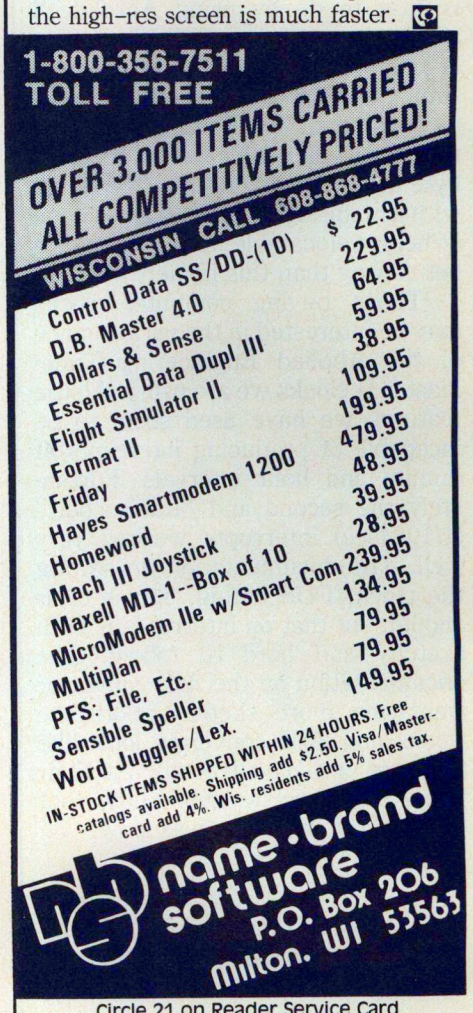
Question

As a computer programming teacher, I have come across a dilemma in the Applevision program on my DOS system master. How did Bob Bishop write text on the high-res screen in the program? The author's name crawls across the screen and the words "Home Sweet Home" are printed on the picture on the wall.

Answer

I don't know exactly how Bob Bishop did that as there are several ways to print text on the high-res screen. There is no simple way such as a POKE or whatnot to allow this. One popular method is to define each character as a standard Applesoft shape (see pp. 92-100 of the old *Applesoft Reference Manual* to find out about shapes). These shapes can be drawn to the screen using the Applesoft DRAW or XDRAW commands. It is quite tedious to define your own set of shapes by hand, and besides there are a number of programs available with "font editors" that easily allow you to create character shapes. Two such programs are Higher Text Plus sold by A.P.P.L.E. for $27.50, and Apple Mechanic by Beagle Bros which retails for $29.95. Both programs contain some redefined character sets and the means to print them on the high-res screen. Additional character sets are sold for each, although the two programs and their character sets are not compatible.

A second method is to define the characters in terms of a matrix of dots and use a special machine language routine to "print" the dots on the screen. An example of this is the HRCG (Hi-Res Character Generator) program included on the Apple DOS Toolkit disk. Using the Animatrix editor program included on the disk, you can create your own special character sets, as well as using the other 21 fonts available on the disk. Then, after loading HRCG into memory, simply printing to the text screen will cause the corresponding character to appear in the equivalent location on the high-res screen. The disadvantage of HRCG and Animatrix is that it is somewhat difficult to define characters bigger than seven dots wide by eight dots high, whereas using shapes the character can be any size. The advantage of the dot matrix method is that printing to the high-res screen is much faster.



Circle 21 on Reader Service Card

Write—A.P.P.L.E.

THE READERS

Timely Interrupts

Dear Editor:

HAVE RUN across a problem while writing an assembly language interrupt handling routine. The routine loads the accumulator with a value placed at an address by an Applied Engineering 8 bit A/D converter and stores it in a table at $8300 (above Ampergraph and below DOS). Storing the data in the first 266 bytes is easy to do by using Absolute,X addressing but how can the high byte be changed? I solved the problem by simply testing for when the X register was equal to $FF and branching to a loop which adds one to the high byte of the base address of the table with a resulting code that has no hope of being relocatable. Surely there is a better way than this homely method.

Those buying computer clocks may be interested in the performance of the Applied Engineering Timemaster II clocks we are using. All the features we have used seem to be incapable of producing interrupts at minute and hour intervals. Fortunately the second and "millisecond" (1/1024sec) interrupts worked very well. The documentation on setting the clock, DOS dating, etc. is clear enough but that on interrupts is both sketchy and hard to follow. The documentation on the A/D converter was even more sketchy. Both the clocks and A/D converter look like they are well constructed. At their price thay are a good buy, but delivery of our clocks was slow.

Stephen Williams
Associate Professor of Biology
Lebanon Valley College
Annville, PA 17003

KoalaPad Reviewer Responds

Dear Mike Oden,

Re: Response to Koala review

Thank you for your response and for pointing out the experiences you have had with the Koala. You are correct in noticing that the early Koalapads had the problem you describe. It was especially noticeable, if while in the draw mode, you lifted the stylus before you released button 0, or if you had used button 1 to lock the draw mode on and then lifted the stylus (even a reduction in pressure being applied could cause stray lines). For that reason, I found that both using the stylus and being conscious of the status of the buttons helped reduce the occurrence of stray lines. However, they were still a problem.

I would like to indicate that my first test pad came right off a local dealer's shelf. I didn't stress in my review that I had also noticed a high frequency of stray lines upon first using this unit. I, at first, thought it was my unfamiliarity with the Koala. I found stray lines to be most prevalent in software other than what is included with the Koala. I didn't mention that the reason I "opened 'er up" was because I was trying to find an adjustment to solve that very problem. Consequently, I spoke to the manufacturer and returned my pad for an exchange unit. The replacement showed no signs of the stray lines except in pre-Koala drawing programs.

During my conversations with both Koala Technologies and Island Graphics (producer of the software), I was able to learn that there were some problems with the early Koalas and that software routines were also added to help filter the RF noise that was associated with the stray lines. Since my replacement Koalapad performed well, and the stray line was reduced to an almost non-existent occurrence, I assumed (shame on me) that the problem had been taken care of.

Just yesterday I tried to recreate the problem you describe. When using the Micro-Illustrator that comes with the pad, I did not experience any stray lines. However, while using another drawing package, I was able to produce the stray line event. It certainly is frustrating to see one's high-res efforts be ruined because of technical errors. Perhaps frequent 'back-ups' or 'touch-ups' are required to alleviate the painful necessity of starting from scratch. As with other means of expression, patience and technique become part of the creativity.

Although the Koala really isn't a replacement for the Graphics Tablet, I still feel there is a great advantage over the joystick for which it is a direct replacement. Compared to the usefulness and performance of most joysticks, I still rate the Koala as Very-Good to Excellent. If the device indeed requires filtering routines, as was indicated to me, I feel they should be included in the manual for the benefit of all.

Alan R. Fischer
Box 425
Hughsonville, NY 12537

Expanding with ALF and Aztec

To the Editor:

It is a useful coincidence that articles of the Manx Aztec "C" compiler under Apple DOS and the review of ALF's 8088 co-processor (*Call-Apple, Feb.,1984*) appeared nearly

simultaneously, for the two subject matters are not entirely independent.

To begin more or less chronologically, I have been using the ALF 8088 co-processor for about a year. Initially my purpose was to reduce computation time on floating point transcendental function intensive programs, which the ALF Formula Transfer Link software admirably does with Applesoft and TASC compiled programs (and with other compilers, I suppose). The manner in which FTL occupies the RAM card also allows the monitor modification which permits lower case display. In addition, ALF supplied a tiny patch to permit the Saturn Card DOS mover to function. These made me a content ALF customer.

Late last year ALF made available, at an affordable price, their version of CPM-86. Although my vision for its application was yet vague, I purchased their CPM-86 and their accessory 128K RAM card. For a brief time I used the additional 128K as a RAM-disk (software supplied by ALF with the RAM card) under DOS which also functioned as advertised. Then I discovered that ALF/CPM-86 was no toy; that the 8088 could address Apple memory and peripherals; and that some programs written for the IBM under CPM-86 (how many I do not know) when reformatted for Apple disks under the standard Apple-CPM format run without further modification!

Having come this far, I talked (not without considerable trepidation) to the Manx Aztec "C" compiler people about getting their IBM/CPM-86 compiler downloaded onto Apple-CPM format disks. I frankly had no confidence it would work but the Alf people did. Harry Suckow at Manx said "no problem" and eventually I received my Aztec "C" compiler. (I have since learned that "no problem" is computerese for 4-6 months, but that is a different matter). I made the permitted back-up copies, organizing what was needed on TWO logically partitioned disks; one disk for the compiler-assembler-editor and another for the linker-library. At that moment the ALF 8088, CPM-86, and Manx's "C" compiler took over my Apple][+. This Apple on my desk is now a computer with unusual versatility and capabilities. I boot up to an 80 column machine with 40 track drives with flashing underline cursor that runs Aztec "C" and the few other CPM-86 programs I have, flawlessly. The ALF RAM card has a slot for an 8087 which I recently filled and I eagerly await the Manx "C"-compiler update that will support Intel's 8087.

There is no question in my mind that any Apple user who wishes to run "C" would be well advised to do it on the ALF 8088 with 128K additional RAM. Running "C" on an Apple under the 6502 processor is very inhibiting and does little to advance the use of "C" (which, in my view, is both powerful and lovely) by Apple owners. For those who wish to take advantage of the Aztec "C" 8088 assembly language interface, there is virtually no limit to what you can do with Apple peripherals, although most of the peripherals are transparently handled by the CPM operating system drivers. Also bear in mind that under the 8088, the Apple][+ has at least 192K of fully addressable memory. The ALF 8088 and Aztec "C" compiler have given my Apple a new lease on life.

A. Wasserman
1610 NW Woodland Dr
Corvalis, OR 97330

Good Mail-Order Service

Dear Editor:

I wish to report to you an example of what I consider exemplary service by a mail-order house which has an extensive catalog of Apple software and peripherals. The company is *3A Computer Products,* of Cheyenne, WY. In February, 1983 I ordered the *Graftrax* chips for my MX-100 printer. Ignorant of the fact that there would be different chips for the MX-80 and the 100, I inadvertently ordered the ones for the MX-80.

Believe it or not, it was not until November that I realized my mistake, when I began trying to print a summary of my tax year. My printer thought it was an MX-80, and wouldn't print past 80 columns! (I made the mistake of first complaining to A.P.P.L.E. that something was wrong with my copy of *The Spreadsheet*; sorry about that.)

I wrote to Ron Maleika at *3A*, and he replaced the chips with the correct ones, free of charge, even though I had owned them 9 months and even though the mistake was mine. You can't beat that kind of service. By the way, his prices, especially for hardware, are very competitive. Would you believe that there are still places in this country where people live 270 miles from the nearest computer store? Mail order is a necessity for people like me. Ron is always willing to provide copies of ads and sometimes even reviews of the merchandise he sells, in advance of the sale.

L.A. Walton
Box 1849
Sidney, Montana 59270

DOS Bugs Persist

Dear Val:

The two DOS 3.3 bugs described in my article "One Down, One to Go" in *Call-A.P.P.L.E. In Depth, Vol. III: All About DOS*, still exist in the latest version of DOS 3.3. Both bugs involve the subroutine at $AE42 (in a 48K DOS) that prints a 3-digit decimal number for the CATALOG function handler.

The first bug, in the subroutine itself, causes the number of sectors in a file to be printed incorrectly if the file contains more than 255 sectors. The second bug, which becomes noticeable only after the first bug is fixed, passes an incorrect Disk Volume number that is too large for the subroutine to print, causing it to print inverse garbage instead of the correct Volume number.

The patches in Listings 5 and 6 in the article will fix these two bugs without taking any more space than the original version of DOS. In the interests of everyone having access to a bug-free DOS, I would like to encourage software developers and publishers to use these patches in their products. My only request is that they let me know if they decide to use the patches, and give me credit in their documentation for being the author. It wouldn't hurt if they would also give A.P.P.L.E. credit for publishing the corrections to DOS by citing the *All About DOS* volume.

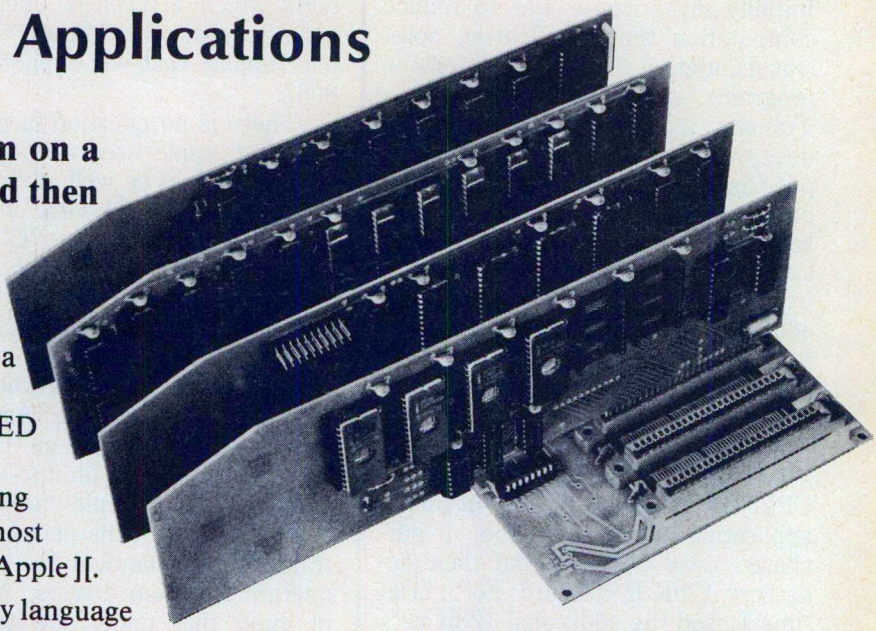
Richard Norling
1217 E Street, S.E.
Washington, D.C. 20003

APPLESEED

T.M.

Apple Compatible Hardware for Dedicated Applications

Design and program a system on a standard Apple computer and then market it with the low-cost APPLESEED system . . .



Designed for use in systems where a full Apple computer may not be necessary or desired, the APPLESEED Bus system retains all the essential features of an Apple system, including mother board compatibility with almost every plug-in board available for the Apple][.

Programs are developed in assembly language on a standard Apple][. Upon completion, a set of EPROM's are blown and inserted into the APPLESEED Bus system, which then takes control of program execution.

You buy only what you need for your application; gone are the built-in keyboard, the consumer-oriented packaging and graphics subsystems. Cost, size and power consumption are substantially reduced while flexibility is greatly increased. Compatibility with a multitude of sophisticated programming and hardware add-ons, widely available for the Apple][, has been fully retained, providing a wide range of options for designing customized systems. (Applesoft PROM's are not included.)

A sampling of APPLESEED Bus components:

CPU BOARD The 6502 has been removed from the mother board and included in this card, which also contains the clock circuitry for the system.

64K RAM BOARD 64K of dynamic RAM with provision for control to be used in the normal Apple][conventions; deleting COOO-CFFF and software control over the DOOO-FFFF area.

EPROM BOARD Jumper selectable for 2716 (EPROM) or 34199 (PROM); blow 2716 assembly language programs and then run them in this EPROM board. Jumpers allow selection of software on/off, initialize on/off and other controls.

UART BOARD Software/hardware select all standard EIA baud rates. The board supports four 6850 UART's which allow four independent input and output EIA lines. Each UART is fully programmable for parity, bit length, baud rate, etc.

UNIVERSAL BOARD The APPLESEED Universal board contains the following: One 2716 socket for an F8 PROM, one 2K × 8 bit RAM (0-7FF), one UART channel for serial communications, 8 parallel INPUT lines (TTL), 8 parallel OUTPUT lines (TTL), device decode for one additional standard slot-dependent board, support for standard Apple][style keyboard, and a power supply connection. Each feature is independent and any combination may be used simultaneously. The Universal board is designed to be used with the APPLESEED CPU board.

TEXT BOARD The TEXT board is designed to add the conventional TEXT (page 1 and page 2) to an APPLESEED system. The board generates composite video to plug into a standard monitor or T.V. modulator. The board provides 40 columns × 24 rows in each page and is not slot dependent.

MOTHER BOARD 10-slot motherboard is fully compatible with most Apple add-on cards currently available.

Please call or write for prices and more detailed information on these and other APPLESEED Bus products.

DOUGLAS ELECTRONICS 718 Marina Blvd. • San Leandro, CA 94577 • (415) 483-8770

Apple][is a registered trademark of APPLE COMPUTER INC.

Circle 17 on Reader Service Card

MORE INFORMATION about the products and services in this issue FREE!

Just circle the number and fill out the card. (Note: card expires three months following month of publication.)

Act Now! Join! A.P.P.L.E.

Apple **P**ugetSound **P**rogram **L**ibrary **E**xchange

The **World's** Largest Apple Computer User Group

Be a part of the over 20,000 A.P.P.L.E. members and you'll receive:

- 12 issues of Call–A.P.P.L.E. magazine. The nation's premier Apple computer publication.
- Substantially reduced prices on sophisticated software, hardware, publications, and services.
- Access to the A.P.P.L.E. closed user group on the **SOURCE.** Excellent for information exchange and program downloading.
- **Call —A.P.P.L.E.** Hotline.

ALL THESE BENEFITS AND MORE FOR ONLY $51!

Detach and mail the Business Reply Application Form. →

READER SERVICE CARD

Free Information on the Items circled

Card expires August 30, 1984 (5-84)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120
121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150

SURVEY

What one computer do you currently use most?

☐ Apple II ☐ Apple //e
☐ None ☐ Other (specify)

What computer (if any) would you likely purchase in the near future?

☐ Apple //e ☐ Apple ///
☐ Lisa ☐ IBM
☐ MacIntosh ☐ Apple //c

What hardware product are you most likely to buy in the next 4 months?

☐ Hard Disk
☐ 5 megabyte
☐ 10 megabyte
☐ 20 megabyte
☐ 8086 board for your Apple (run IBM P/C software)
☐ Apple keyboard (increase capability

Name _____

Company _____

Address _____

City _____

State _____ Zip _____

Call —A.P.P.L.E. Reader's Choice

Help us help you. By filling out and returning this card, marked with your preferences, you can guide us in selecting material tailored to your needs. Just place the numbers from 1 to 10, in order of preference, in the appropriate boxes. Please do not choose more than 10 features. A.P.P.L.E. members who include their membership number, will have their names placed in a drawing for $100 credit toward any A.P.P.L.E. product. Drawing will be held July 17, 1984. The top scoring author will also receive a $100 cash bonus.

☐ **A First Look at Apple's NEW IIc**
Dave Lingwood

☐ **About Robots: An Account of Costs**
Michael Thyng

☐ **HEROs and Apples**
John Covington and Burt Webb

☐ **Artificial Intelligence: The Myth and the Reality**
Jim Wicker, M.A.

☐ **Using an Apple to Speak English to a Robot?**
Susan K. Lynch, RB Robot Corp.

☐ **The Graphics Toolkit: Part 2**
Randi J. Rost

☐ **The Gibson Light Pen: An "Etch-a-Sketch" Graphics System for the Apple**
Lee Webster

☐ **The AppleMouse][**
John Cox

☐ **The Apple][Circuit Description**
Leonard H. Anderson

☐ **Adding a TYPE Command to ProDOS**
Cecil Fretwell

☐ **ProDOS — Pros and Cons**
Cornelis Bongers

☐ **THE SPREADSHEET SHEET: Data Analysis (Part 6)**
Jeffrey M. Jacques, Ph.D.

☐ **TOPO — A Tutor and Tool for Your Classroom**
Harvey Barnett

☐ **The Turtle as a Helpful Robot**
Nola Sheffer

☐ **Touch Sensors for Screen Turtles**
Jeff Sandys

☐ **A.P.P.L.E. Doctor**
Bruce F. Field

☐ **Write—A.P.P.L.E.**
THE READERS

☐ **Tomorrow's Apples Today**

☐ **Unclassifieds**

☐ **Member Billboard**
Don Elman

☐ **The INNER CIRCUIT**
Don Elman

Name _____ Address _____ Memb. # _____

A.P.P.L.E. employees ineligible for prizes. Void where prohibited by law.

☐ Please send me an A.P.P.L.E. Apple-cation and information.

$25.00 First time fee
26.00 First year dues
$51.00 Total

☐ $51.00 payment enclosed.

☐ Mastercharge/Visa

MasterCard # _____ Exp. Date _____

VISA # _____ Exp. Date _____

Name _____

Address _____

City _____ State _____

Zip _____ Phone _____

Apple PugetSound Program Library Exchange
The WORLD'S largest Apple Computer user group

A.P.P.L.E., 21246 68th Ave. S., Kent, WA 98032
(206) 872-2245

First Class mail for **Call —A.P.P.L.E.** add $15. Otherwise third class mailed free as part of membership in U.S.A.

Additional Airmail postage required for foreign memberships and shipments.

Additional membership postage:
- Canada, West Indies, Central America, Columbia, Venezuela and Mexico — Add $15.00 ($66.00 total)
- All S. American countries (except above), Europe, and N. Africa — Add $21.00 ($72.00 total)
- Asia, Mid East, S. Africa, Australia and Pacific Islands — Add $27.00 ($78.00 total)

All orders must be prepaid in U.S. funds only.

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD
FIRST CLASS PERMIT NO. 217 CLINTON, IA

POSTAGE WILL BE PAID BY ADDRESSEE

Call-A.P.P.L.E.
P.O. Box 1993
Clinton, IA 52735

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD
FIRST CLASS PERMIT NO. 32 KENT, WA

POSTAGE WILL BE PAID BY ADDRESSEE

Call-A.P.P.L.E.
21246 68th Ave. S.
Kent, WA 98032

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD
FIRST CLASS PERMIT NO. 32 KENT, WA

POSTAGE WILL BE PAID BY ADDRESSEE

A.P.P.L.E.
21246 68th Ave. S.
Kent, WA 98032

cont'd from p. 9

The case is much thinner than on the earlier models, but when an Apple staffer stood on it it survived! There is a protective grey canvas travel case for it, with room for power pack, modulator, some supplies, and even the flat panel display.

SUPPORTING CAST

The manuals and instructional diskettes set a new standard. The books are clear, giving just enough information for new owners, with good graphics and explanatory pictures and diagrams.

The "Interactive Owner's Guide" takes the new user through the fundamentals, gives demos of typical applications, and provides a taste of Applesoft programming and Logo. Double-sided diskettes contain the following sides:

- An Introduction (keyboard, common operations)
- The Apple at Play (games, a loan program)
- The Apple at Work (simulation of AppleWorks word processing, database and spreadsheet)
- The Inside Story (shows how computer works)
- Exploring Apple Logo (simulated in Pascal!)
- Getting Down to BASIC (introductory concepts)
- System Utilities (the real stuff: formats either ProDOS, 3.3 or Pascal diskettes, converts, etc.— an update of the ProDOS system utilities; also configures the serial ports).

And, what do you know? The manual also discusses user groups and magazines, including A.P.P.L.E., as information resources for new owners.

EXTRAS

There is a new line of peripherals with the same look as the *IIc*:

- Monitor *IIc*: a compact green-screen for under $200, with optional tilt stand ($39).
- *IIc* Mouse: unit and MousePaint, $99.
- External ½ height disk drive, under $350.
- Scribe printer: a completely new unit, using "thermal transfer" process on plain paper in either black or color! Priced at $299. This unit is aimed at the home user with needs for a limited volume of printing. The color quality, even onto transparency stock, is good enough for executive presentations, however. See the accompanying article.

Of course, other Apple peripherals, as well as those of other suppliers, will fit: serial printers, the Apple plotter, 300 and 300/1200 baud modems, etc.

SUMMING UP

The *IIc* isn't designed for expansion. No slots mean no extra goodies — at least until the manufacturers adapt to those DIN serial connectors and the RGB and disk sockets. No currently-available hard disk will fit it. We doubt that anyone will care, however. Few home users will ever get beyond the resources provided by the serial, RGB and disk connectors. Current *II* owners will latch onto the *IIc* for its portability and compatibility with the software they now have.

How about the 65C02 itself? The issue sounds like the 8080-Z80 problem. When the Z80 came out there was already so much 8080 equipment around that few commercial programs took advantage of the improved op codes of the Z80. The only recourse for software is to put two versions of the program on the disk, then load the appropriate one. We doubt there are enough speed or compactness advantages to the 65C02 to make it worth the trouble. Of course, software could always just include a 65C02 for *IIe* owners to plug into their computers (you can't do this to a][+, though). Note that A.P.P.L.E. now has a *IIc* version of Big Mac.

Apple's planning to sell tons of these things. The Dallas plant can be cranked up eventually to turn out a *IIc* every seven seconds, with half the labor required to make a *IIe*.

We think Apple WILL sell them by the ton. There are the many thousands of folks who have been in the market for a good home computer, but couldn't quite cope with the hassles of external drives, slots, jargon, and configuration. The whole marketing effort is aimed squarely at these people.

These new users may have to wave over the heads of crowds of current][and *IIe* owners, however. I'm betting that a large proportion of sales will go to present owners, at least in the first few months. For one thing, the *IIc* is just too darned cute to ignore. It fits perfectly as the second Apple for the business or professional person: easy to take home at night, easy to lug around on the road, the perfect fit when you need to hide out in North Shoeshoe to get the program or report finished, the answer for in-the-field computing for scientific or other data gathering.

Yes, the San Francisco event is correctly named. The *IIc* will help insure, as did the *IIe*, that the Apple][family will be around for a long time — maybe even forever.

IIc SPECIFICATIONS

- Weight: 7½ lb.
- Dimensions: 2½" x 12" x 11½".
- Processor: 65C02, 8 bit.
- RAM: 64K, plus 64K auxiliary memory bank.
- ROM: 16K, enhanced monitor, Applesoft, 80 column, 2 super serial cards, disk controller, mouse. *IIe* compatible.
- *IIc* Signature bytes: $FBB3 = 06, $FBC0 = $00.
- Keyboard: 63 key upper/lower case; same as on *IIe*, two switch-selectable configurations (Qwerty and Dvorak in the U.S.).
- Controls & indicators: RESET, 40/80, keyboard configuration, disk drive on LED (red), power on / low-voltage LED (green).
- Disk drive: built in (S6,D1), 1/2 height "pop out" style. External (S6,D2) optional. Two drive maximum. Cannot use DOS 3.1 or 3.2 (conversion to 3.3 supplied on Utilities disk). Speed adjustment is through the access hole in the bottom of the case.
- Speaker: built-in, external headphone jack with volume control.
- Slot assignment equivalents to the *IIe*: all built-in.
  1 Super Serial (for printer)
  2 Super Serial (for modem)
  3 Auxiliary memory 80 column card
  6 Disk controller.
- Power supply: external power pack: 15vdc, 1.2 Amp. Rear-panel DIN connector accepts +9vdc — +18vdc. Power-on LED blinks if DC in is less than 9vdc.

NIBBLE EXPRESS III
YOUR APPLE* WILL LOVE IT!

TABLE OF CONTENTS

Turtle Graphics	8
Applesoft Line Cruncher	14
TRAC Income System	19
MAMA	25
Amper Free Space Catalog	26
Machine Language Editor	27
Apple CALENDAR	36
Peaceful Coexistence	43
Apple Slugger	43
Reverse Key	45
Amper Find	45
Bond Manager	47
DOS 3 + 2	53
Amper Speed	57
Apple Darts	61
MicroCalc	63
Variable Cruncher	72
Life	77
Apple Flash	80
Checker	81
GO- Greeting Program	83
Quick Sort	86
Applesoft Line Editor	90
Othello	96
SPRINT	99
Disk Dump	102
Hi-Res Colors	103
Apple Record Command System	104
Apple Art Gallery	114
Game I/O Expmt	119
Super Keypad	123
Disk Commander	125
Compare Applesoft	132
Amp-L-Soft	136
Electronic Message Center	152
The Shape	155
Disk Map	159
Apple Bowl Football	161
Apple Scroller	167
Lower Case Letters	168
Recipe Box	170
Disk Doctor	178
Quasar II	182
DOS Command Entry	189
MLE Driver	189

NOW AVAILABLE!

NIBBLE EXPRESS is an Anthology of the Major Articles and Programs appearing in Volume 3 of NIBBLE Magazine . . . nearly $400 worth of Programs for your Apple II, ACE, or other Applesoft-compatible computer!

NIBBLE EXPRESS III contains *up-to-date enhancements, KeyPerfect Tables,* and the *Best of NIBBLE!*

Even if you have all 8 issues of NIBBLE in 1982, you'll want the Express to have updated program listings in one convenient package! It's a MUST for your Library!

NIBBLE EXPRESS III is a bargain at $17.95 plus $1.75 Postage/Handling ($2.50 outside the U.S.).

YOUR APPLE WILL LOVE IT!
(And so will you!)

NIBBLE
P.O. Box 325
Lincoln, MA 01773

Yes! I want to reserve NIBBLE EXPRESS Vol. III for my Library! Here's my ☐ Check ☐ Money Order for $17.95 plus $1.75 postage/handling. (Outside U.S. add $2.75 for postage/handling).

☐ Also send me NIBBLE EXPRESS Vol. I at $12.95 Plus postage/handling (see above).

☐ Send me NIBBLE EXPRESS VOL. II at $14.95 plus postage/handling (see above).

Master Card & Visa Accepted

Card # _____ Expires: _____

Signature: _____

Telephone: _____

Please Print Clearly

Name: _____

Address: _____

City _____ State _____ Zip _____

Your check or money order must accompany your order to qualify.
Outside U.S.: Checks must be drawn on a U.S. Bank.

*Apple is a registered trademark of Apple Computer Inc.
*ACE is a trademark of Franklin Computer Co.

Circle 44 on Reader Service Card

- Rear-panel:
  - On-off switch
  - Power (DIN male)
  - Serial (DIN female) for printer
  - External drive (DB19)
  - Video out (RCA phono)
  - RGB - video out (DB15, matches RF modulator)
  - Serial (DIN female) for modem
  - Mouse/joystick/paddle (DB9)
  - Carrying handle.
- Package includes:
  - *IIc* computer
  - RF modulator, TV connector switch, 12 ft. video cable
  - AC power pack, 6 ft. AC cord, 6 ft. 15vdc cable
  - Manuals
  - Introductory diskettes, utilities diskette. ProDOS operating system.

List price: $1,295.

The Scribe Printer

Color printing for under $300? How did Apple do it?

First, you need to understand what "thermal transfer" is — how the printer gets whatever color from the ribbon to the paper. First off, it is NOT the same as the old "thermal" technology that required special heat-sensitive paper. In the Scribe the paper isn't heat-sensitive, the RIBBON is. The paper can be anything: bond, photocopy paper, standard pin-feed, etc.

The Scribe print head consists of 24 dot heads, arranged vertically. Each head is a resistive heating unit. The print head pushes the ribbon just up to the paper. When the printer "fires" a dot, the heat heats up — very quickly. The heat transfers a dot of the ribbon color onto the paper.

The black ribbon looks like a mylar "multi-strike" daisywheel ribbon. Looks like, but it isn't. The active material on the ribbon is more akin to crayon than to ink. Heat from the head dots melts a dot-worth of the ribbon onto the paper.

The process is faster than it sounds. Scribe will tool along at 80 characters per second in draft mode, and 50 characters per second in letter-quality mode. Quick enough for a low-volume printer. That translates to 25 lines per minute in draft mode, and 20 lines per minute in letter quality mode.

How do they do color? The same way. The color ribbon is made up of alternating 10-inch sections of the three colors magenta, cyan and yellow. That is, for ten inches, the whole ribbon is one color, followed by an equal length in another color, etc. This means the printer needs to be able to tell which color of ribbon is "up" now. It does this by an optical sensor reading clear bands on the ribbon.

Obviously, if you plan to print a long section in one color the printer will have to hop over the two intervening wrong-color strips when it reaches the end of the correct color. In short, this is not a system designed for economical printing of lots of text in color. Again, we need to remember the market for which this printer was designed: home users and occasional "executive" use.

There are only three color controls: ESC K n, where "n" equals 1 for magenta, 2 for cyan, and 3 for yellow. What if you want to make other colors by combining these? Software must do the job, printing a line in one color, then returning with no linefeed to print the next color(s).

Color printout is vivid and well-saturated. Both color and black density are greater than on some of the low-cost (under $1,000) color ink jet printers we've seen.

Apple is providing no HIRES color- dump software. At least two companies are introducing programs to do the job, however.

For text processing you snap the black ribbon back in. The ribbon cartridges pop out and in easily. Both the color and black ribbons are one-pass units. When the ribbon runs out the printer stops and flashes its LEDs to tell you what has happened. A black ribbon is good for about 80,000 characters of printing.

The black ribbon will sell for about $7.00, color for about $9.00.

Paper handling is smoother than we've seen anywhere else. Single sheets feed easily and squarely. Pin feed paper is pulled through by the tractor (as opposed to the push-through used on the ImageWriter). This means you waste the first sheet, as with many other printers (e.g., Epson). It is done this way to get better paper movement accuracy.

The commands supported in text mode are a sub-set of those used on the ImageWriter. Enhanced and double-width work. Smaller characters are printed without vertical paper movement for sub and super scripts. Underlining is low enough that it doesn't print over a sub script. The printer simply ignores those ImageWriter commands it doesn't support.

What's missing is the noise. Scribe makes no more noise than the un-lamentedly defunct Silentype. It can co-exist peaceably in the family room, the library, or the meeting room.

The only other thing missing is 12 pitch. There simply wasn't room in the ROM to pack this in along with the other features. Apple decided that 17 pitch was more important than 12, since you need 17 to be able to print a 132 column spreadsheet model on 8½ inch paper.

The Scribe is designed to work with more than just the *IIc*, of course. Mac and Lisa can cope with it, as could any][or *IIe* equipped with a serial card. A simple pop-out cover in the back panel hides eight DIP switches. These come configured for use with the *IIc*, but can be changed for use with different computers or serial interfaces.

The printer comes with a plastic stand that will hold just about a ribbon's worth of paper beneath. The RS232 serial and power connectors emerge from the back sides of the printer case, not from the rear. This lets the printer operate close to a back wall or desk partition. The "footprint" and operating space were designed to be small — with an eye toward the limited computer desk space usually available in the home.

In sum, this is a versatile, yet very "targeted" printer. The target market is clearly the home user who produces a low volume of printout. The surprisingly vivid color output, even on transparencies, will also fit nicely with the manager who needs to make presentations every now and then. You wouldn't want to use this unit for high-volume business correspondence, however. At $299, though, the Scribe will follow you home very nicely, thank you.

CHICAGO
NEW YORK

FORMERLY APPLEFEST & PC'83

THE PERSONAL COMPUTER USERFEST

Plan now to attend the Personal Computer Userfests, the largest events ever . . . for Apple and IBM PC users.

Userfest brings together two of the largest, most successful shows ever conceived for personal computer users: Applefest and PC'83. Now that Apple and IBM can run each other's software, and with so many products adapted for both systems, the two shows merged beautifully.

At Userfest you'll see—and try out—all of the newest state-of-the-art products for your Apple, IBM PC or work-alike. Each Show has hundreds of displays and exhibits, and thousands and thousands of products including innovative new software, power peripherals, accessories, support services, books and publications. Products to help you explore the full potential of your computer for office, home and school applications.

Userfest features all the major makes of Apple and IBM computer compatibles. In fact, it's the largest display of these products, and biggest gathering of IBM and Apple experts, ever assembled in either city. Hence, you can learn more in two days at Userfest than you could in months of visiting computer stores or reading trade journals.

And best of all, everything on display at Userfest is for sale, usually at special show prices, so you can save hundreds, even thousands of dollars by making your purchases at the Show

So don't miss the Personal Computer Userfest when it comes to Chicago and New York in 1984. It's a once-only opportunity.

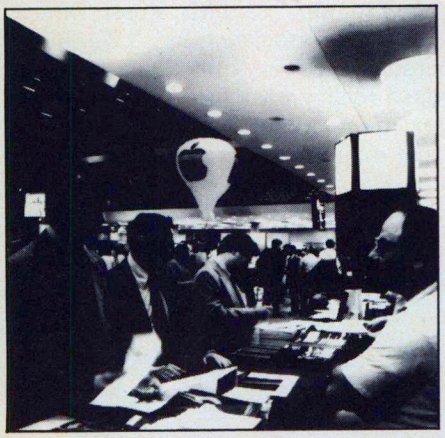
Order your tickets in advance and avoid long lines. Admission is $10.00 for a one-day ticket, or $20.00 for four days. Children's tickets (under 10 years of age) are $4.00 and $8.00. If you need hotel accommodations and/or airline reservations, check the line on the Advance Ticket form.

Circle 19 on Reader Service Card

FEATURING IBM, APPLE & WORK-ALIKE COMPUTERS & COMPATIBLES

THE PERSONAL COMPUTER

USERFEST



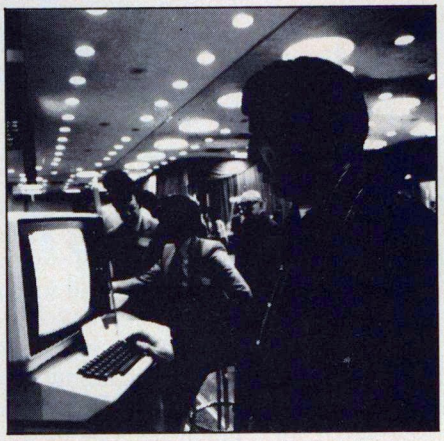
USERFEST/ CHICAGO

Thursday-Sunday
May 3-6, 1984
10:00AM to 5:00PM daily
O'Hare Exposition Center
9291 West Bryn Mawr
Rosemont, Illinois
(next to Chicago's O'Hare Airport)

USERFEST/ NEW YORK

Thursday-Sunday
September 20-23, 1984
Madison Square Garden
10:00AM to 5:00PM daily

For information about exhibiting at the Personal Computer Userfests, call or write Northeast Expositions, 822 Boylston Street, Chestnut Hill, Mass 02167. Tel: 617-739-2000.



For hotel information call or write Trade Show Department, Fox Travel, P.O. Box 498, Waltham, Mass 02254. Tel: 617-890-1770 or 800-225-8410 ext. 314.

Userfest (formerly known as Applefest and PC'83) is produced by Northeast Expositions, 822 Boylston Street, Chestnut Hill, Mass 02167.

ADVANCE TICKET ORDER FORM

Mail this form (or a facsimile) with full payment to Northeast Expositions, 822 Boylston Street, Chestnut Hill, Mass 02167. Tel: 617-739-2000. No ticket orders accepted 14 days or nearer to each Show. Your tickets will be mailed one month prior to the Show. Sorry, no telephone or credit card orders please.

Name: _____

Company (if any): _____

Address: _____

City: _____ State: _____ Zip: _____

Tel: Day (____) _____ Evening (____) _____

Enclosed is full payment for:

_____ quantity adult one-day tickets @ $10.00 each

_____ quantity adult four-day tickets @ $20.00 each

_____

use this line for children's ticket orders

☐ Check here if you need hotel and/or airline information

Tomorrow's Apples Today

NOTE: These products are presumed to be state-of-the-art. Call -A.P.P.L.E. does not necessarily endorse products published in this column as we have not tested them.

NEW APPLE DUAL-DISK DRIVE AVAILABLE FOR APPLE *II* COMPUTERS

Cupertino, CA — A new floppy disk drive unit which contains two half-high, 140-kilobyte drives side-by-side in a single case was introduced today by Apple Computer. Called Duodisk, the new drive offers compact styling, improved technology and full compatibility with Apple *II* software at a lower price than two separate Disk II drives.

Duodisk sets a new standard of economy for Apple *II* users, meeting the dual-drive configuration needed for many software programs, simple back-up and ease of use. This new drive is suited to the professional, small business, office, higher-level education and developer markets where two drives are required for most applications.

Duodisk is exactly the width of the Apple *II* and can sit conveniently between the computer and monitor. Its case was designed to complement the Apple *II* and the Monitor *II*, thereby creating an integrated system look.

Technical improvements include a new disk eject mechanism and an advanced head positioning mechanism which gives a more precise reading of half-tracks.

Duodisk comes with a controller card that connects it to any Apple *II*, Apple *II+* or Apple *IIe*. It is operationally identical to the Disk *II* floppy disk drive and runs all Apple *II* software.

Price and Availability:

Duodisk is available now from all authorized Apple dealers worldwide and carries a U.S. suggested retail price of $795. For comparison, it would cost $940 to purchase two Disk *II*'s and a controller card.

Duodisk is covered by Apple's 90-day warranty. Apple's extended service plan, AppleCare(TM), is available for coverage beyond the first 90 days.

TURTLE TOT MIGRATES TO NORTH AMERICA

Somerville, MA — Turtle Tot, a new personal robot, is now available in North America. Built in Australia by Flexible Systems of Hobart, Tasmania, pre-production Turtle Tots were demonstrated at the Comdex Las Vegas and Canadian Computer shows. Turtle Tot can move, draw, turn, blink its "eyes" and feel its surroundings with touch sensors. With an optional speech package, Turtle Tots can even talk.

"With a retail price of $299, Turtle Tot is already selling at an unprecedented rate," said Mr. Rossi, Director of Marketing at Harvard Associates, Inc. "We think the idea of a truly personal, yet very affectionate Turtle robot has now become a reality with Tot. With features found only on larger, more expensive robots, Tot is available at a price that makes people want to take him home!"

Turtle Tot is controlled via an RS-232 serial interface allowing it to be used with most microcomputers. All Turtle Tots come fully assembled and tested with easy-to-read instructions and a 10-color pen set.

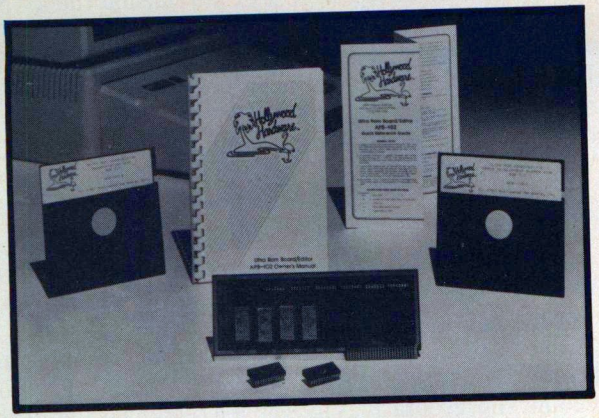
The Turtle concept was originally used to introduce Logo, the high-level programming language popular in the educational computing community. Mr. Rossi noted that the Turtle Tot represents "a new generation of Turtles, one in which even pre-school children are able to control a sophisticated robot. With its new low price, every classroom can now have its own personal pet," he concluded.

For more information about Turtle Tots write: *Harvard Associates Inc., 260 Beacon St., Somerville, MA 02143* or call *(617) 492-0660.*

NEW SOFTWARE KIT OFFERS SCHOOLS BIG SAVINGS ON APPLE LOGO

Cupertino, CA — The Apple Logo Classroom Kit, a new product from Apple Computer, allows schools to equip 10 Apple *II* personal computers with the popular

Would You Like to Make Programming Just Plain **Easier?** Easier to Learn— Easier To Do— and 3 Times **Faster?!**



The Hollywood Hardware Firmware Enhancement System

Why You Need It:

We take all the enhancements that transform the Apple from a novelty to a powerful instrument, and hook them into the operating system, installed on their own FIRMWARE card. NO disk loading, NO loss of available memory space, NO interface with other programs. The system never need be removed - it is unhooked with two keystrokes, rehooked with four. ALWAYS IN THE MACHINE, NEVER IN THE WAY.

The Response: We Get a Lot of Nice Comments:

"...Excellent product, flawless.." Phil Daley

"...Hollywood Hardware has developed an elegant solution...it makes editing an Applesoft program all it should have been in the first place, and then some."... SOFTALK review

"...The best thing for the Apple since the disk" Edward Decker

The Basic System Includes: **$189.00**

a) **The ULTRA-ROM BOARD (APB102A)** Installs 32K of firmware space in any slot of your II, II+, //e. Requires no program memory.
"Nice looking, high quality hardware."... Robert Zedelis

b) **GPLE 4.2:** Enhanced version of the original editor that ALLOWS good programming practice. GLOBAL SEARCH; automatic EDIT or REPLACE; INSERT, DELETE, FIND, ZAP; CASE CONVERT and RESTORE variables, or characters. We guarantee you will never want to be without it.
FUNCTIONS: Sits between the keyboard and the rest of the Apple - so any key can double as a FUNCTION KEY that generates a whole command string. Over 60 of the most used are provided as MACROs. System users never type "CATALOG, D1", or SAVE <filename>". Just type TWO keys: ESC 1, or ESC CTRL-S.
CUSTOMIZE and save your own macros!
"The most powerful program tool I have..." Robert Wilson
"If you program, and haven't used a line editor, get one right away..." Micro Magazine review

c) **FMS: The FIRMWARE MANAGEMENT SYSTEM** Finds and enables the desired routine with automatic bank switching. New ROMS are recognized automatically.
"...The integrated memory managment system is one of the best features of the board." Mark Simonson, Beagle Brothers.

d) **APU-1:** Over 25 UTILITIES. Invoked by the "&" key: **Ultra fast searches, Garbaged program recovery**. Instant numerical **conversions**, Graphics state controls. Language extensions like PRINT USING and IF, THEN, ELSE are included, and provided in disk form for portability.
"...P.S. love it! Can hardly wait for more!" Paul Fox

Optional Additions Include:

e) **NEW!! APU-2: supeRENUMBER, multiHIDE, multiMERGE, VARIABLE CROSS REFERENCE, AUTONUMBER.** NEW programs by Paul Johnson. Far more powerful and mistake proof than other versions...and you don't destroy your own program loading them in from disk. They are always at hand. **$35.00**

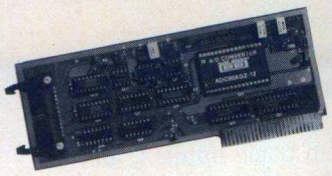
f) **NEW!! COPY ZAP by Glen Bredon - author of BIG MAC.** SMART DISK COPIER, and ZAP utilities written for our system. Also copies 40 track disks! Hit "&COPY" and watch! **$39.95**

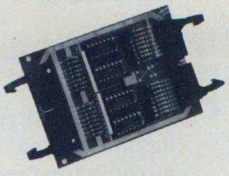
g) **NEW!! ROM DEVELOPEMENT SYSTEM:** Program your own firmware! Includes software, documentation and emulation RAM. **$99.00**

NEW NEW NEW NEW

12 Bit 16 Channel Data Acquistion

12 BIT 16 CH. A/D SIGNAL CONDITIONING





AD16B $299.95

- ★ IRQ or NMI INTERRUPT
- ★ PRECISION SAMPLE & HOLD
- ★ EXTERNAL START CONVERT
- ★ HIGH SPEED - 25,000 CONV/SEC
- ★ 7 VOLTAGE RANGES
- ★ PRECISION REFERENCE OUT
- ★ SUCCESSIVE APPROX. CONVERTOR
- ★ COMPLETE SOFTWARE INCL. DEMOSCOPE

A16G $79.95

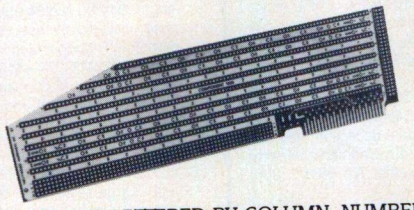
- ★ BOLT-ON FITS IN APPLE
- ★ 16 OP AMPS
- ★ EACH CHANNEL SEPARATE GAIN & FILTERING

A8D from $149.95

- ★ 2 to 8 CHANNELS
- ★ TRUE DIFFERENTIAL INPUTS
- ★ BOLT-ON FITS IN APPLE
- ★ GAIN JUMPER EACH CHANNEL
- ★ GAINS OF 1, 10, 100, 1000

EXPANDED LINE

PRO-1 PROTOTYPING BOARD



THE BIGGEST
THE BEST
$29.95

- ★ ALL IC'S LETTERED BY COLUMN, NUMBERED BY ROW
- ★ ACCEPTS ALL STANDARD SOCKETS - UP TO 52 (16 PIN)
- ★ NUMBERED INPUT & OUPUT PINS ACCEPT I.D.S. CONNECTOR
- ★ ±5 AND ±12 SUPPLIES HANDY & WITH SPACE FOR FILTERING
- ★ GOLD PLATED EDGE CONNECTOR, PREMIUM BLACK BOARD

48 Line Parallel I/O PD48

$249.95

W/50 COND. RIBBON CABLE

- ★ 25 FILTERED LINES IN
- ★ 23 BUFFERED LINES OUT
- ★ INTERRUPT INPUT
- ★ 4 TIMERS

ADDITIONAL FINE PRODUCTS:

★ **MACROTECH MEMORY EXPANSION**
DISCULATOR 64K — $379.95
DISCULATOR 128K — 459.95

★ **ANCHOR AUTOMATION MODEMS**
VM-1 VOLKSMODEM 300 BAUD — $64.95
VM-1 CABLE — 9.95
SIGNALMAN MARK IL 300 BAUD — 79.95
MARK VII, 300 BAUD, AUTO — 109.95
MARK XII SMART MODEM — 299.95
300/1200 BAUD AUTO ANSWER
BELL 212, RS 232, HAYES* COMPATABLE

★ **HOLLYWOOD HARDWARE T-SHIRT**
SM, MED, LG, XLG — 12.00

*TRADEMARK HAYES MICROCOMPUTER PRODUCTS

Hollywood Hardware 6842 VALJEAN AVENUE, VAN NUYS, CA 91406 (818) 989-1204

*GPLE © 1983 NEIL KONZEN, SOLD UNDER LICENSE FROM SYNERGISTIC SOFTWARE / APPLE IS A REGISTERED TRADEMARK OF APPLE COMPUTERS INC.

educational programming language, Apple Logo, for substantially less than the cost of buying individual Apple Logo packages.

The kit provides:

- one standard Apple Logo package that includes a master and back-up disk, an introductory manual and a tutorial manual.
- eight additional Apple Logo program disks.
- ten Apple Logo mini-reference cards.
- one Apple Logo Tool Kit set that includes a Sample Programs disk, a Tool Kit disk containing Apple logo product enhancements and Tool Kit manual (the disks can be copied) and,
- a form for ordering additional Apple Logo manuals.

The Apple Logo Classroom Kit carries a suggested retail price of $400, which represents more than $100 savings compared to the cost of purchasing the kit components separately.

Apple Logo is a computer programming language that helps students of all ages grasp fundamental programming concepts quickly. By providing a programming environment that encourages experimentation, Apple Logo allows users to develop language and problem-solving skills and to explore complex programming ideas and geometric and design principles.

Apple Logo runs on any Apple *II*, Apple *II* Plus or Apple *IIe* personal computer with at least 64 kilobytes of internal memory.

The Apple Logo Classroom Kit is available now through authorized Apple dealers and educational account representatives in Apple's regional sales offices.

Unclassifieds

OFFICE HAZARDS: AWARENESS AND CONTROL — Conference May 31—June 2 in Seattle that will examine health problems associated with VDTs. Contact: Jan Schwert, Environmental Health, SC-34, University of Washington, Seattle, WA 98195 (206) 543-1069. Fee: $ 250.

USE OF MICROCOMPUTERS IN OCCUPATIONAL SAFETY AND HEALTH — Course June 13-15 in Seattle that will use the Apple //e for safety and health data collection, analysis and, retrieval. Contact: Jan Schwert, Environmental Health, SC-34, University of Washington, Seattle, WA 98195 (206) 543-1069. Fee: $ 375.

//e ABSOLUTE RESET to the "old" monitor. Get back in total control. User transparent til activated. No hole blasting. Passes diagnostics. Instructions & sourcecode on disk, plus "must have" bonus book $19.95. Old Reset, % SYNERGETICS, Box 1300-AP, Thatcher, AZ 85552 (602) 428-4073. VISA accepted.

APPLE SHAPE EDITOR-Everything you ever really wanted or needed but couldn't buy this cheap! Why spend a bundle for a user's NIGHTMARE when there's a powerful and user-friendly editor, shuffler, compressor, appender, merger, and displayer with ON-LINE HELP for only $19.95? Clean up tables created by expensive programs! Includes popular type fonts and complete documentation. Apple][+,//e DOS 3.3. Send check or M.O. today. Ask for MIRACLEWORKER,COMPUTER ODYSSEY, Box 10434, Pittsburgh, PA 15234. PA residents add 6%.

EAMON ADVENTURE SPECIAL- Set 1 of 19 adventures plus dungeon designer disk. Set 2 of 19 adventures plus utilities disk. Set 3 of 19 adventures plus advanced dungeon designer disk. Any set for $35. All 60 plus monster utility disk for $95. Postage included. Send a SASE for complete list and make your own set of adventures for $40. Satisfaction guaranteed or your money cheerfully refunded. M. Dube, P.O. Box 423C, 8019 Windy Lane NW, Massilon, OH 44646.

PARENTS GUIDE TO POISON PREVENTION — 'In Depth Poisoning' is the first of a series of health educational programs. Written by Richard Voight,M.D., this program is for anyone who cares for children. It comes with a 3 week money-back guarantee if not satisfied. Apple][+,//e 48K. $39.95 + $2.50 shipping. Iowa residents add 4% tax, Health Ed Software, P.O. Box 1209, Fairfield , Iowa 52556. Upcoming topic 'Headaches'.

RENT SOFTWARE-$5 monthly. No membership fee. Send large SASE to: Preowned Software, 4229 Baker Ave NW, Seattle, WA 98107.

dBASE USER wants to trade ideas, applications, command files, printouts, reports for home and general use. For details send SASE to : Greenwald, Box 401, Wheatley Heights, NY 11798.

EAMON ADVENTURES-Eamon starter kit of 18 adventures (Set 1) on 9 disks for $35. Set 2 of 18 for $35. Set 3 of 14 adventures and utilities for $30. First 36 for $60. All 50 for $75. Sample disk $5. 100 ss/sd generic disks with envelopes for $140: Jeff Bianco 7210 North Mercer Way, Mercer Island, WA 98040.

EXTENDED DISK DRIVE CABLE (with connectors). Four-foot length: $14.95 (2/$24.95)—custom lengths available. Complete installation instructions included. Dealer inquiries welcome. Ribbons: Gemini/Okidata—$2.75 (12/$29);MX-80—$6(6/$32);MX-100—$10(6/$55). VISA/MC (include exp. date), check or M.O. Please add $2 shipping/handling per order. The Soft Spot, Inc. 2816 SE Stark St., Portland, OR 97214.

GOLDCOAST COMPUTER RENTAL CLUB, FREE MEMBERSHIP. Large selection of commercial programs available for rent, buy, sell, trade, software, firmware, hardware, and publications. For free details, send to: 2037 Wink Ct., Owensboro, KY 42301.

SPRING HOUSECLEANING CLEARANCE-Take advantage of my lack of space! Nibble Magazine, all issues from V.1,N.6 (late 1980) to date, except V.4,N.7. Creative Computing from V.3,N.1 (Jan.,1977) to date. Magazines are in very good condition; you may bid on portions of the sets if you wish. Also, A.L.S. Z-Card II (CP/M), used 20 hours, with registration card. Games:SSI Computer Baseball, Bill Budge's Raster Blaster, the original Ultima, Wizardry I, II, and III. Graphics: TME Arcade Board, Synetix SuperSprite Board with StarSprite Software vols. 1,2,3. All reasonable offers will be considered. If you would like a reply in case your bid is unsuccessful, please enclose SASE. Thank you! Ken Sherwood, 117 N. 25th St., Reading, PA 19606. All offers must be received by June 30.

STOCK MARKET SOFTWARE WANTED- Will pay for programs that use the RTR data disks. Especially wanted are USER routines for the DJMA (Apple II version), i.e.,Wells Wilder, %A, Magic T, Cycle Analysis, etc. Send information to W.Wolfe, Box 238, New York, NY 10470.

700 PROGRAMS & MORE: Over 700 programs available for business, education, and games for the][and //e $8. Discounted peripheral cards. Details: Reliant, P.O. Box 33610, Sheungwan, Hong Kong.

TIME ZONE AND MYSTERY HOUSE SOLUTIONS: includes maps and instructions. Time Zone $4. Mystery House $2. Both for $5. Contact Duane Allman, 12805 W. Eden Trail, New Berlin, WI 53151.

MULTI-LINGUAL software for the Apple][+ or //e. FREE CATALOG. Le Professeur, 959 N.W. 53rd St., Ft. Lauderdale, FL 33309.(305) 771-6498.

EDUCATIONAL SOFTWARE: Send for free price list of educational software for the Apple. The Country Store, P.O. Box 2131-Ced., Littleton, CO 80161.(303)779-0883.

KEYNOTER.Specifically designed to hold those valuable notes you need at the keyboard. We offer 2 types: flat overlay which fits around the keyboard or detachable clear plastic book which holds 6 3x5 cards. Please specify which KEYNOTER + Apple //e or][+ when ordering. Send $9.95 + $1.50 shipping (CA res. add sales tax) to: Geisler Enterprises, 20333 Old Town Rd., Star Rte. #2, Tehachapi, CA 93561.

NOTCH-0 PUNCH: Want to write on the back of your disk? Use our self-aligning Notch-o punch to add a write-protect notch. $8.95 + $1.05 shipping (CA res. add sales tax). Geisler Enterprises, 20333 Old Town Rd., Star Rte. #2, Tehachapi, CA 93561.

AIR NAV WORKSHOP ([+,//e), fully-customizable, navigate any area, adjustable winds, automatic flight path replay, much more,$40.Check, MC/V, UPS-COD. SPACE-TIME ASSOCIATES, 2039-CA Country Club Drive, Manchester, NH 03102.(603)625-1094. Free info.

OBJ.APWRT][F-Detailed,complete, and thorough disassembly script includes full details on customizing Applewriter //e and capturing your own source code. Two unlocked, crammed-full, double-sided diskettes plus a free bonus book for only $29.95. SOURCECODE, %Synergetics, Box 1300-AP, Thatcher, AZ 85552.(602) 428-4073. VISA accepted.

ALMOST WHOLESALE — Save BIG BUCKS on almost all Hardware & Software needs. Usually 20% to 45% below retail. Monthly specials. Send self-addressed stamped envelope for latest price list, or contact us to request quotes on specific items. COMPUTER PRODUCTS UNLIMITED, Apple Dept., 11116 East 46th Ave., Spokane, WA 99206. (509) 922-1346 (eves). CompuServe (73075,613). Source (TCG249).

MICROMOVER quickly transfers files between Apple disks over a phone line, without a conversion first. Data Compression algorithm. DOS 3.2 and 3.3 Hayes Micromodem II and acoustic modems accommodated. Only one copy required, as it transfers itself to distant end. Cost is $50 from DAB Systems, 9116 Rockefeller Lane, Springfield, VA 22153.

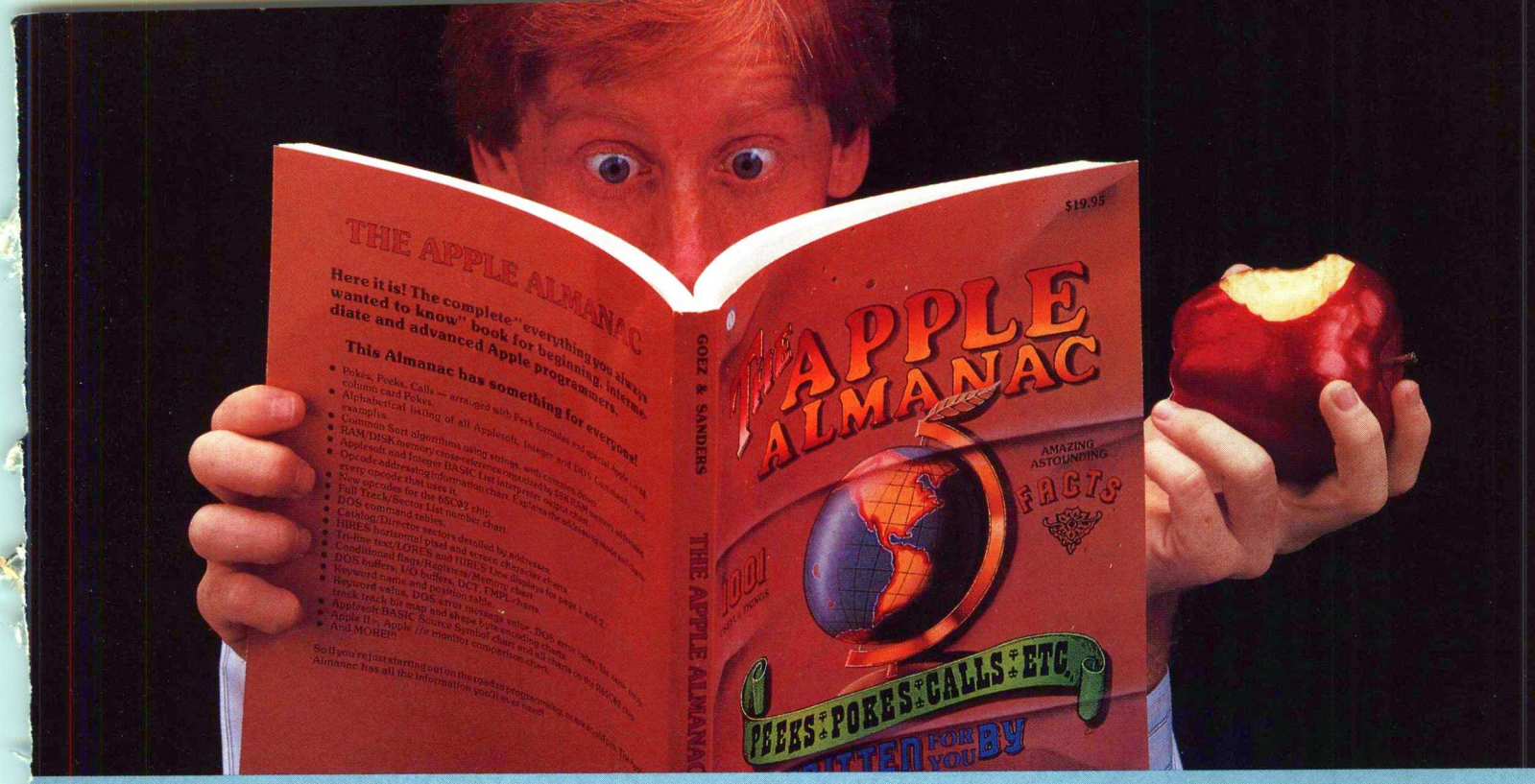
INTERFACE CARDS FOR WHOLESALE PRICES: 16K RAM Card $49, Z80 Card $39, 128K RAM Card $149, 8088 Card $199, 6809 Card $139, 80 Column Card $59, and 30 more different kinds of cards for special prices. HUNTER ELECTRONIC CO., P.O. BOX 53-130, TAIPEI,TAIWAN, REP. OF CHINA.

DISK O' UTILITIES! Thirteen useful APPLE utilities on disk for about a buck each! A Bargain! Send $13.95 to: BROADWAY SOFTWARE, Suite 136, 660 Amsterdam, New York, NY 10025 (VISA/MC).

3 UTILITIES FOR $50! LOCKER: Protect your program from copiers. FINDER: Find any commands, for fast editing. SHRINKER: Rewrites program; saves memory! To order, mail $50 to DAN SCHMIDT 15841 Leadwell St., Van Nuys, CA 91406.

COMPARE:	THE FILER	TO PFS
Search 300 files	1 sec	3 min.
# of files on disk	4000	1000
PRICE	$50	$125

To order mail $50 to DAN SCHMIDT 15841, Leadwell St., Van Nuys, CA 91406.



ALL YOUR APPLE ANSWERS IN ONE BOOK!

THE APPLE ALMANAC
By Eric E. Goez and William B. Sanders Ph.D.

You know the scenario. A late night. The Apple* II, II+ or IIe is humming. Piles of newsletters, charts, and assorted books surrounding you. All for a single algorithm! Wouldn't it be nice to have ONE source to find that ASCII, opcode, PEEK/POKE, and DOS chart you've been searching for?

THE APPLE ALMANAC is that ONE source! Easy to use and packed with valuable programming information, the ALMANAC is arranged so you can find what you want INSTANTLY. So whether you're a beginner or an old pro, THE APPLE ALMANAC is that *one-of-a-kind* reference book you've been waiting for!

DATAMOST™

The most out of our minds™

Datamost, Inc., 20660 Nordhoff St., Chatsworth, CA. 91311. (818) 709-1202

*Apple is a trademark of Apple Computer.

Circle 2 on Reader Service Card

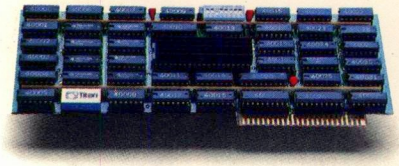
How to make your Apple run 3½ times faster.

Just plug in the Accelerator IIe™ and watch your Apple® II, II Plus or IIe take off.

Imagine running VisiCalc,® DB Master,® Applesoft, Apple Fortran or Pascal without long delays. With the Titan Accelerator IIe, your Apple II, II Plus or IIe runs these programs a true 3½ times faster. The Accelerator IIe also runs your Franklin, Basis and other Apple II compatibles 3½ times faster. This faster computer response time means less waiting for you and an increase in your productivity.

This Titan exclusive has its own fast 6502 processor, 64K of high-speed memory, and built-in fast language card. It is transparent to your software and is hardware compatible with most standard *peripherals*. And you can turn if off from the keyboard to run your Apple at regular speed.

Find out how to run faster and increase your productivity without breathing hard. For information on the Accelerator IIe and other Titan products, see your computer dealer or contact:



Titan TECHNOLOGIES, INC.

310 West Ann Street, Ann Arbor, MI 48104
Telephone (313) 662-8542

Apple is a registered trademark of Apple Computer Inc.
VisiCalc is a registered trademark of VisiCorp.
DB Master is a registered trademark of Stoneware, Inc.
Accelerator IIe is a trademark of Titan Technologies, Inc.

Sales and Marketing by The MARKETING RESOURCE GROUP, Fountain Valley, CA.

Circle 3 on Reader Service Card